UNIVERSITI TUN HUSSEIN ONN MALAYSIA

STATUS CONFIRMATION FOR THESIS
DOCTOR OF PHILOSOPHY

A GUIDED ARTIFICIAL BEE COLONY(GABC) HEURISTIC FOR PERMUTATION FLOWSHOP SCHEDULING PROBLEM (PFSP)

ACADEMIC SESSION: 2020/2021

I, **NOOR AZIZAH BINTI SIDEK,** agree to allow Thesis to be kept at the Library under the following terms:

1. This Thesis is the property of the Universiti Tun Hussein Onn Malaysia.
2. The library has the right to make copies for educational purposes only.
3. The library is allowed to make copies of this Thesis for educational exchange between higher educational institutions.
4. The library is allowed to make available full text access of the digital copy via the internet by Universiti Tun Hussein Onn Malaysia in downloadable format provided that the Thesis is not subject to an embargo. Should an embargo be in place, the digital copy will only be made available as set out above once the embargo has expired.
5. ** Please Mark (√)

☐	CONFIDENTIAL	(Contains information of high security or of great importance to Malaysia as STIPULATED under the OFFICIAL SECRET ACT 1972) *Title and Abstract only*
☐	RESTRICTED	(Contains restricted information as determined by the organization/institution where research was conducted)- *Title, Abstract and Introduction only*
☐	EMBARGO	______ (date) ______ (date)
√	FREE ACCESS	

Approved by,

(WRITER'S SIGNATURE) (SUPERVISOR'S SIGNATURE)

Permanent Address:

NO 80, JALAN JAYA 3/4,
TAMAN PAGOH JAYA,
84600 PAGOH,
MUAR JOHOR

Date : 11/01/2021 Date: 12/01/2021

NOTE: ** If this Thesis is classified as CONFIDENTIAL or RESTRICTED, please attach the letter from the relevant authority/organization stating reasons and duration for such classification.

This thesis has been examined on date 15th November 2020

and is sufficient in fulfilling the scope and quality for the purpose of awarding the Degree of Doctor of Philosophy.

Chairperson:

PROF. DR. ERWEEN BIN ABD RAHIM

Faculty of Mechanical and Manufacturing Engineering

Universiti Tun Hussein Onn Malaysia

Examiners:

PROF. TS. DR. BADRUL BIN OMAR

Faculty of Mechanical and Manufacturing Engineering

Universiti Tun Hussein Onn Malaysia

ASSOCIATE PROF. DR. MUHAMMAD HAFIDZ FAZLI BIN MD FAUADI

Faculty of Manufacturing Engineering

Universiti Teknikal Malaysia Melaka

A GUIDED ARTIFICIAL BEE COLONY (GABC) HEURISTIC FOR PERMUTATION FLOWSHOP SCHEDULING PROBLEM (PFSP)

NOOR AZIZAH BINTI SIDEK

A thesis submitted in
fulfilment of the requirement for the award of the
Doctor of Philosophy

Faculty of Mechanical and Manufacturing Engineering
Universiti Tun Hussein Onn Malaysia

JAN 2021

I hereby declare that the work in this project report is my own except for quotations and summaries which have been duly acknowledged

Student : ……………………………..……..

NOOR AZIZAH BINTI SIDEK

Date : 11/01/2021 ………………………………………

Supervisor : ………………………………………

ASSOCIATE PROF. DR. SH SALLEH BIN SH AHMAD

Co Supervisor : ……………………………………….

ASSOCIATE PROF. DR. MOHD ZAMANI BIN NGALI

This thesis is dedicated to

My parents, Allahyarham Sidek bin Hashim and Zainab binti Mohmad

My husband, Azli bin Nawawi

and to

My three beautiful children,

Muhammad Naufal,

Naurah Elena,

and

Naurah Elana

ACKNOWLEDGEMENT

Normally, research often consists of several sessions of failures. But one cannot deny the fact that failure is the greatest teacher of all and provides an exceptional platform for the author to learn and strive. However, on top of that, the author must acknowledge and express all the gratitude to **ALLAH SWT,** the most gracious and merciful, for blessing, protecting and guiding the author throughout the period. Additionally, the author would like to thank all the precious people that provide the greatest support with significant strength and stay behind the author, very closely and lovingly, to support the author for bouncing back to face all those failures.

The author would also like to express her deepest gratitude to the main supervisor, Assoc. Prof. Dr Sh Salleh bin Sh Ahmad for his continuous support, a tremendous level of patience and admirable guidance that enable the author to have an excellent atmosphere for doing research. Besides, the author would also like to thank her co-supervisor, Assoc. Prof. Dr Mohd Zamani bin Ngali for giving her the precious additional guidance and the best exposure during his journey for PhD and undergraduate project.

The author would also like to express the deepest gratitude to her parents, Allahyarham Sidek bin Hashim and Zainab Binti Mohmad, who give her the utmost motivation and support for overcoming all obstacles during her learning period.

The greatest thank is also due to his loving and supportive spouse, Azli bin Nawawi, as he is everything to her and always be by her side during the ups and downs. He never gives up and keeps supporting her no matter what happens. Not forgotten, her beautiful children, Muhammad Naufal, Naurah Elena and Naurah Elana, whose act as the main pillar of her strength and determination. They inspired her in so many ways. Each child inspires her in a very meaningful way and their presence aids to build a stronger and wiser version of her. May **ALLAH SWT** bless us all.

ABSTRACT

Flowshop is the most common production system in the industry, and there are many documented efforts to improve the performance of the flowshop. The range spreads from the usage of heuristics to metaheuristics, and one of the promising methods is NEH (Nawaz, Enscore & Ham) heuristics. This study aims to improve NEH, using an enhanced version of Artificial Bee Colony (ABC) algorithm because the original one has the problem of slow converge speed. As a result, this study will propose a mechanism to improve the convergence speed of ABC because faster convergence speed is the ability to find high-quality results in lesser iterations compared to others. The study clusters the Employed Bees (EB) and Onlooker Bees (OB) into several groups: Total Greedy, Semi Greedy and Non-Greedy. Upon completion, the study selected the Total Greedy (3+0+0) because of the leading performance in makespan value (performance indicator), and the author used it for the rest of this study. This study proposed two variants of the guided initial ABC or Guided Artificial Bee Colony (GABC) with one variant (NEH-based ABC), employing the concept of NEH and the second variant (GABC), employing the concept of NEH and First Job Sequence Arrangement Method. The study experimented according to ten datasets of Taillard benchmark and divided the experiments into several categories and the experiments run every data for several iterations, and for each dataset, there are 20 replications. This study compared the performance of NEH, ABC, NEH-based ABC and GABC, which also act as the validation process. Based on the results, ABC produced inconsistent results for a significant amount of times and interestingly, GABC, NEH-based ABC and ABC produced 68.75%, 63.33% and 0.01% results that are better than NEH, respectively. The data also shows that GABC is 37.9% better than its variant. Finally, the author can conclude that this study demonstrated the slow convergence issue of ABC.

ABSTRAK

Flowshop merupakan sistem pembuatan yang paling kerap digunakan di industry dan terdapat pelbagai usaha yang telah didokumenkan bagi meningkatkan keupayaan *flowshop.* Kaedah yang digunakan juga pelbagai, bermula dari heuristik ke heurustik-meta dan kaedah yang paling popular adalah heuristik Nawaz, Enscore dan Ham (NEH). Kajian ini bertujuan untuk meningkat potensi NEH, menggunakan algoritma *Artificial Bee Colony (ABC)* yang telah dipertingkat kerana versi asal ABC adalah agak perlahan untuk mendapatkan penyelesaian berkualiti tinggi. Oleh itu, kajian ini bertujuan untuk menjadikan ABC berupaya menjana penyelesaian yang baik dalam jumlah iterasi yang lebih rendah. Kajian ini membahagikan *Employed Bees (EB)* dan *Onlooker Bees (OB)* kepada beberapa kluster: *Total Greedy, Semi Greedy* dan *Non-Greedy*. Setelah ujikaji dijalankan, *Total Greedy* dipilih kerana menjana masa siap yang paling baik dan penulis menggunakan pilihan ini di keseluruhan projek. Kajian ini mengetengahkan dua varian *guided initial ABC* atau *GABC,* dimana varian pertama (ABC berdasarkan NEH), menggunpakai konsep NEH dalam pengoperasian. Varian kedua pula (*GABC*), menggunapakai konsep NEH dan teknik *first job sequence arrangement*. Ujikaji dijalankan berdasarkan data dari Piawaian Taillard dan membahagikan ujikaji kepada beberapa kategori. Ujikaji juga dijalankan dengan beberapa nilai iterasi, dan setiap set data perlu menjalani 20 sesi ulangan. Kajian ini membandingkan prestasi NEH, ABC, ABC berdasarkan NEH dan GABC, dan peringkat ini juga dinamakan proses validasi. Berdasarkan hasil ujikaji, ABC didapati menjana keputusan yang tidak konsisten dalam kekerapan yang tinggi. Agak menarik kerana keputusan yang dihasilkan oleh GABC, ABC berdasarkan NEH dan ABC, 68.75%, 63.33% dan 0.01% lebih baik dari NEH secara keseluruhan. Data ujikaji juga menunjukkan GABC adalah 37.9% lebih baik berbanding variannya iaitu ABC berdasarkan NEH. Akhir sekali, dapat disimpulkan bahawa kajian ini menunjukkan demonstrasi berkaitan isu *convergence* yang perlahan oleh algoritma ABC.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

ABC	-	Artificial Bee Colony
ACO	-	Ant Colony Optimization
BMSA	-	Bi-objective Multi-start Simulated Annealing algorithm
CDS	-	Campbell, Dudek, and Smith
DE	-	Differential Evolution
DLS	-	Dynamic Local Search
DNEH	-	Discrete Nawaz Enscore Ham heuristic
DPFSP	-	Distributed Permutation Flowshop Scheduling Problem
EA	-	Evolutionary Algorithm
EC	-	Evolutionary Computation
ES	-	Evolutionary Strategy
GA	-	Genetic Algorithm
GABC	-	Guided Artificial Bee Colony
GRASP	-	Greedy Randomized Adaptive Search Procedure
IG	-	Iterated Greedy
ILS	-	Iterated Local Search
MA	-	Memetic Algorithm
NC	-	Nature Computing
NEH	-	Nawaz Enscore Ham
NP	-	Non-Deterministic Polynomial Time
PF	-	Profile-fitting
PFH	-	Profile-fitting Heuristics
PFSP	-	Permutation Flowshop Scheduling Problem

PSO	-	Particle Swarm Optimization
RTS	-	Reactive Tabu Search
SA	-	Simulated Annealing
SPV	-	Smallest Position Value
SS	-	Scatter Search
TA	-	Threshold Accepting
TC	-	Trial Counter
TS	-	Tabu Search
VNS	-	Variable Neighbourhood Search
WBS	-	Work Breakdown Structure
FSS	-	Flowshop Scheduling
FSP	-	Flowshop Scheduling Problem
ACS	-	Ant Colony System
MOACSA	-	Multi-objective Ant Colony System Algorithm
HAMC	-	Hybrid Algorithms for Multicriterion
CR (MC)	-	Chandrasekharan Rajendran (Multicriterion)
AIS	-	Artificial Immune System
FSH	-	Hybrid Flowshop
CPU	-	Computer Processing Unit
MCFMS	-	Multi-Cell Flexible Manufacturing System
SEAIS	-	Self-Evolving Artificial Immune System
AIS-IG	-	Artificial Immune System Iterated Greedy
DABC	-	Discrete Artificial Bee Colony
hDDE	-	Hybrid Discrete Differential Evolution
EDA	-	Estimation Distribution Algorithm
hGLS	-	Hybrid Genetic Local Search
EDD	-	Earliest Due Date
LSL	-	Last Slack Time

OSL	-	Overall Slack Time
IABC	-	Improved Artificial Bee Colony
IBFSP	-	Flowshop Scheduling Problem with Intermediate Buffers
EB	-	Employed Bee
OB	-	Onlooker Bee
SB	-	Scout Bees
HABC	-	Hybrid Pareto-based Artificial Bee Colony
JSP	-	Job Shop Scheduling
DPSO	-	Discrete Particle Swarm Optimization
APRI	-	Aspartate Aminotransferase to Platelet Ratio Index
PSOVNS	-	Particle Swarm Optimization Variable Neighbourhood Search
PSOMA	-	Particle Swarm Optimization based Memetic Algorithm
VBA	-	Visual Basic for Application
EBOB	-	Employed Bee and Onlooker Bee
SDST	-	Sequence Dependent Setup Times
EM	-	Electromagnetism-like mechanism
MILP	-	Mixed Integer Linear Programming
tsGLS	-	Genetic Algorithm Hybridized with Tabu Search
vIG_DE	-	Variable iterated greedy algorithm (IG) with differential evolution
NEHedd	-	Algorithm of Nawaz, Enscore and Ham with EDD rule for preprocessing
MOGA	-	Energy-efficient Genetic Algorithm
MOGALS	-	A variant of MOGA
MASC	-	Memetic algorithm with novel semi-constructive crossover and mutation operators
HPSO	-	A hybrid algorithm based on particle swarm optimization
LPT	-	Longest-Processing-Time-first algorithm
FCFS	-	First Come First Serve

FIFO	-	First In First Out
LCFS	-	Last Come, First Served
SPT	-	Shortest Processing Time

CHAPTER 1

INTRODUCTION

1.1 Challenges of Flowshop Scheduling

Fast, customised, and efficient are the benefits of a flowshop, which Fan & Winley (2008) described as a fabrication facility used to manufacture one or more types of products. The focus of the flowshop is to make products with the highest quality in the shortest time. The flowshop system has a fixed job flow, and this criterion is quite useful for a high quantity production system (Bandyopadhyay, 2019; Fan & Winley, 2008; Park, 1976). The flowshop is also the goal of a lean system, and it is easier to control compared to other system types such as Job Shop and Project Shop (Pinedo, 2012). The most significant advantage of the flowshop is repeatability, and it has a linear machines arrangement, and the system processes each job at every machine, with only in one direction (Groover, 2016). Permutation flowshop scheduling problem (PFSP) is one of the most discussed problems in FSP. The purpose of PFSP is to find the best processing (job) sequence to satisfy the optimization requirement. In most of the literature, the objective PFSP optimization is the makespan, which is considered as an act to minimize the total completion time for the production processes (Liu & Liu, 2013; Naderi & Ruiz, 2014; Sidek & Bareduan, 2014).

The main challenge of using the flowshop is to find the best job sequence that will minimise the makespan (Gupta & Stafford, 2006). It is crucial because if the manufacturer can churn out a product faster, it will enter the market early, and the manufacturer will have a higher chance of gaining the market share (Wolde, Berhan, & Jilcha, 2018). Managing the production process with the need for the scheduling processes take part is not easy and is considered as NP-hard (Garey, Johnson, & Sethi, 2008a; Kan & Kan, 2011). Since it is NP-hard, the heuristic method is the best way of solution for the flowshop problem (Liu & Liu, 2013). After Johnson introduced the flowshop problem (Johnson, 1954), it has been a magnet for research. Previous studies proposed a few problems solving approaches using a heuristic method such as Palmer heuristic (Palmer, 1965), Gupta heuristic (Gupta, 1971), CDS (Campbell, Dudek, & Smith, 1970) heuristic, and NEH heuristic (Nawaz, Enscore, & Ham, 1983a). Nawaz, Enscore and Ham introduced the NEH heuristic around three decades ago, but the scheduling community still consider it as superior due to its ability to minimize the makespan of Permutation Flowshop Scheduling Problems (PFSP) effectively. The majority still prefer NEH for its priority order method, and this helps to achieve near-optimum solutions (Framinan, Leisten, & Rajendran, 2003).

NEH also has shortcomings, and more research appears to improve NEH such as done by Taillard (1990) which applies Tabu Search. A study by Kalczynski & Kamburowski (2008a) focuses on improving NEH by proposing a new priority order with a simple tie-breaking method. Another effort for improving NEH is made by Ruiz & Stützle (2007) by selecting the proper solutions to put into NEH using the iterated greedy solution.

NEH is powerful, but it does not utilize the random selection criteria, and the ability to use the random selection of solutions will result in more possibilities of better solutions (Emmons & Vairaktarakis, 2013). From this point of view, the effort to utilize the metaheuristics seems to be next step, with the hope of finding a method that is superior to NEH. Kouki, Guenaoui, and Jemni (2016) focused on the parameter tuning of the Genetic Algorithm (GA) for solving PFSP, and the study managed to churn out a better version of GA. Wang and Yin (2013)also discussed the parameter tuning of the GA and the parameters in consideration are the number of population, probability of crossover and probability of mutation. The parameter tuning process managed to produce a better version of GA. Nowicki & Smutnicki (1996) proposed an algorithm based on the tabu search technique with the application of a block of jobs notion. Another effort made using tabu search was by Gao, Chen, and Deng (2013) by proposing a tabu search algorithm for solving Distributed permutation flowshop scheduling problem (DPFSP). This algorithm exploited a novel tabu search strategy and enhanced the local search method, and it performed better than GA.

NEH is very effective for solving PFSP and some research utilizes the concept of NEH to make their algorithm perform better. Kurdi (2020) proposed a memetic algorithm (MA) combined with a novel semi-constructive crossover and mutation operators (MASC) to solve PFSP. The MASC used in the research is a combination of genetic algorithm (GA), simulated annealing (SA), and NEH algorithm. Öztop, Tasgetiren, Eliiyi, Pan, and Kandiller (2020) proposed a novel PFH_NEH(x) which is the combination of profile-lifting (PF) constructive heuristic and NEH algorithm. The main reason for the team to choose NEH is because it is a simple and effective heuristics for solving PSFP. Wang, Gao, Li, Li, and Tasgetiren (2020) proposed a hybrid version of the whale swarm algorithm with DNEH. In the study, DNEH is used to optimize the makespan's initial solution objective.

Lin and Ying (2013) proposed a bi-objective multi-start simulated annealing algorithm (BMSA) to solve the PFSP by minimizing the makespan. The researchers tested the algorithm using the Taillard benchmark, and as a result, the BMSA managed to contribute 64% of the solutions in the non-dominated front. Osman and Potts (1989) proposed the simulated annealing method to optimize PFSP and experimented with problems of up to 20 machines and 100 jobs. From the study, the researchers found that simulated annealing produced slightly better results compared to NEH in some cases.

When it comes to the ant colony algorithm, Ahmadizar (2012) proposed a novel ACO algorithm which initializes the pheromone trails based on the initial sequence. The algorithm also employs a local search to improve the solution quality, and it managed to perform better than other variants of ant colony optimization in the literature at that time. Mirabi (2011) utilized the ACO algorithm on the sequence dependent PFSP with modifications to the approach for computing pheromone values and the employment of a local search. Meanwhile, Tasgetiren, Liang, Sevkli, and Gencyilmaz (2007) managed to utilize a modified Particle Swarm Optimization (PSO) algorithm for solving PFSP. Modifications made to the PSO algorithm are the application of a heuristic rule named smallest position value (SPV), genetic algorithm (GA), random keys for sequencing and optimization, and the variable neighbourhood search (VNS). In 2004, the same researchers embedded the VNS in the PSO algorithm for solving PFSP, and it produced competitive results. For the total flowtime and makespan criteria (2007 version), the modified PSO managed to improve the solutions generated by other competitive algorithms with 63% and 24.4%, respectively (Tasgetiren et al., 2007).

Tasgetiren, Pan, Suganthan, and Chen, (2010) proposed the Discrete Artificial Bee Colony (DABC) algorithm with the inclusion of the iterated greedy (IG) and iterated local search (ILS) algorithm. The researchers compared the algorithm to the estimation distribution algorithm (EDA) and hybrid genetic local search (hGLS), and it managed to improve 43 out of 60 solutions. In 2011, the same research team proposed the DABC with hybrid differential evolution algorithms, and the results were quite competitive (Tasgetiren, Pan, Suganthan, & Chen, 2011). Liu and Liu (2013) proposed a hybrid discrete artificial bee colony (HDABC) that utilised Greedy Randomized Adaptive Search Procedure (GRASP) inspired by NEH heuristics. HDABC also used discrete operators and algorithms to generate new solutions and it applied the local search to find the best one. From the literature, the author can conclude that the efforts to improve the ABC algorithm in solving PFSP in an ongoing journey. There will always be some aspects of the algorithm that can be enhanced.

Artificial Bee Colony (ABC) algorithm is preferred in this study because it has fewer parameters to control compared to other population-based algorithms. ABC algorithm is also a robust, fast to converge, and flexible type of algorithm (Anam, 2017; Ayan & Kiliç, 2012). ABC is also known to perform better than Genetic Algorithm, Differential Equation (DE) and Particle Swarm Optimization (PSO) (Karaboga & Basturk, 2008; Karaboga & Akay, 2009; Karaboga & Basturk, 2007; Sulaiman, Mohamad-Saleh, & Ghani Abro, 2013). Other than that, ABC is proven to produce good results in optimizing permutation flowshop scheduling (Marinakis, Marinaki, & Matsatsinis, 2009; Tasgetiren et al., 2010).

1.2 Problem Statements

Lately, the ABC algorithm has attracted several researchers in studying its capability for solving manufacturing scheduling problems. Chong, Sivakumar, Low, and Gay, (2006a) used the ABC methodology to construct schedules for job shop problems; Pan, Fatih Tasgetiren, Suganthan, and Chua (2011) proposed a discrete ABC algorithm to solve lot-streaming flowshop scheduling while (Tasgetiren, Pan, Suganthan, & Chen,

2010) reported the ABC application for permutation flowshop scheduling problem (PFSP) using total flowtime criterion. Hakli and Kiran (2020) proposed three search equations for the employed bees and another three for onlooker bees to improve the slow convergence problem. Wang, Shi, and Wang (2020) proposed a formula for location update into ABC to improve the optimization's iterative process. The group also integrate the scout bee phase with a beta distribution function to prevent the ABC to fall easily into local extremum. All these researchers have identified the weakness of ABC is that it is always slow to converge. As such, it will be useful to recommend the future works to be focused directly on establishing mechanisms to help the ABC converge with a low number of iterations.

The original ABC algorithm is known to be slow to converge (Banharnsakun, Achalakul, & Sirinaovakul, 2011; Luo, Wang, & Xiao, 2013; Sulaiman et al., 2013), and when it comes to PFSP, this drawback will delay the process of getting high-quality makespan values. The algorithm works in the iteration mode, which means that it will repeat the same process to find a better solution as the iteration goes. As the iteration proceeds, the candidate solutions tend to get closer to the desired location and this is when the convergence occurs (Gao & Cao, 2012; Trelea, 2003). In ABC, the researchers claimed that the convergence speed is quite slow, and the algorithm will take some time to get the desired solution (Chen, Daud, Zhou, & Elisha Nyamasvisva, 2019; Shubham Gupta & Deep, 2020; Zhang, 2020). The exploitation and exploration activities in the algorithm are said to be well balanced but the convergence speed is still wide open for improvement and ABC is easy to fall into local optimum (Banharnsakun et al., 2011; Gupta & Deep, 2020).

Based on the above literature review, the author can conclude that it is beneficial to explore the ABC algorithm further in the context of solving scheduling and other optimization problems. To improve the ABC effectiveness, an additional mechanism must be added to the current algorithms to help it converge to the desired performance measures. This study aims towards developing a new, improved ABC algorithm consisting of additional converging mechanisms that will make it converge more effectively. To achieve the mentioned goal and based on the trends in the previous related works, researchers tend to assimilate the concept of the leading heuristics, NEH in the development of their algorithm. The common practice is to employ the NEH arrangement in their proposal and hybridizing the NEH concept in their algorithms. However, the practice of utilizing the first job sequence method is never reported and for that reason, this study aims to use the method to improve the performance of ABC in solving PFSP. The whole process is known as a new Guided Artificial Bee Colony (GABC) heuristic specifically intended for solving PFSP.

1.3 Research Objectives

This study embarks on the following objectives:

i. To determine the effectiveness of the ABC algorithm using different bee characteristics in optimizing PFSP.
ii. To improve a new heuristic identified as Guided Artificial Bee Colony (GABC) heuristic to solve PFSP.
iii. To assess the convergence performance of the GABC heuristic against the normal ABC algorithm method.

1.4 Research Scopes

This study consisted of several scopes:

i. This study focuses on permutation flowshop scheduling problems (PFSP).
ii. This study focuses on developing an appropriate algorithm representing the clustered behaviour that is useful to improve the scheduling solution from the normal ABC method. In the behaviour clustering, the bees are grouped into similar behaviour in the context of trial counter number and location of bees.
iii. Visual Basic for Application (VBA) coding embedded within Microsoft Excel is used to develop the GABC program to evaluate flowshop scheduling solutions.
iv. This study utilizes the iteration limits from 500 to 5000 iterations. The number of EB and OB is set between 5 to 15 and the swarm size is limited to 10 until 30.

1.5 Thesis Organization

This chapter presents a brief introduction about the challenges in PFSP, and many efforts by previous researchers to improve the system. The author also highlighted the weaknesses of the ABC algorithm to expose the need to work on a solution. This chapter also covers the objectives and scopes of the study, and the author ended the introduction chapter with the organization of the thesis.

In the next chapter, the study uncovers some theories and significant contributions related to production scheduling, permutation flowshop scheduling problems, metaheuristics used in solving PFSP, and ABC algorithm. This chapter is a strong base for the author to develop the research methodology.

The third chapter covers the general research methodology for this study. In this chapter, the author explained the processes for developing the research methodology.

The fourth chapter focuses on the contribution of the study and the steps to develop the novelty. The author explained the procedure to develop the Guided Artificial Bee Colony (GABC) algorithm.

In the fifth chapter, the study covers the details about the performance measurement of GABC. This chapter also documented the results and discussion for this study as well as the validation process.

Finally, the final chapter provides a conclusion for this study and some recommendations for future works. Moreover, this section also highlighted the contributions made by this study.

.

CHAPTER 2

LITERATURE REVIEW

This chapter focuses on the fundamental theories and knowledge related to the research topics such as scheduling, PFSP, optimization methods, and ABC algorithm. This chapter starts with the broader scope of scheduling and the author will narrow down the scope to the application of the nature-inspired algorithm in solving PFSP.

2.1 Introduction to Scheduling

Scheduling is a decision-making process used in the manufacturing and services industries. The main objective of scheduling applications in both industries is to minimize the completion time or makespan (Lapierre & Ruiz, 2007).

Scheduling is also known as a plan made to fulfil a particular objective. Other elements included in the plan are the time frame and activities of a particular project or process. Additionally, scheduling is useful for assigning workers for a particular job when the project is ongoing. Interestingly, the scheduling process can also help to optimize a system to make it work with better efficiency (Md Fauadi & Murata, 2010). This process is feasible by performing a well-executed scheduling process. From here, the author can claim that a scheduling system can improve the output of a particular system. In most cases, scheduling is useful for determining the duration needed to complete an activity, the completion time, and the predecessor activities (Kerzner, 2013; Lapierre & Ruiz, 2007; Pinedo, 2012).

Similarly, scheduling is also known as a way to manage a particular project as it provides the guidance and pathway that need to complete a project. Scheduling also contains the milestones for completing a particular project (Ballard, 2000). Scheduling is also useful for improving productivity, reducing delivery cost and utilizing the manufacturing resources optimally (Md Fauadi & Murata, 2010).

In some industries, scheduling is very crucial for transportation and distribution settings, and it will also assist most industries in fulfilling their customers' demands while minimizing the total setup time. Moreover, proper scheduling can maximize the throughput of a particular production enterprise. This step can be done by improving the equipment utilization, and from here, scheduling can minimize the number of idle times and setup times. Interestingly, scheduling is also able to minimize the amount of work needed to achieve specific goals, and it is also a useful tool to minimize delays in the production process because the delays in production are equal to the low reliability of the company (Pinedo, 2012). Typical applications of scheduling are 1) Procurement and production, 2) Transportation and distribution, and 3) Information processing and applications (Sauer, 1999).

The practitioners change the plan and control production over time due to customer requirements and environment and technology improvements. Therefore effective manufacturing control processes based on planning and scheduling represent the key to the success of a manufacturing company (Vollmann, William, Whybark, & Jacobs, 2005). Scheduling as a decision process represents an essential tool to determine the sequential order of activities identified by the work breakdown structure (WBS), as part of the project scope (Baldwin & Bordoli, 2014). To do that, the practitioners must perform a well-executed scheduling process. From here, the author can conclude that a scheduling system can improve the output of a particular system.

2.2 Production Scheduling

Production scheduling is known as the process of arranging, controlling, and optimizing all the production budget timescale for every phase of a business project. It has robust and complicated precedence relationships between the manufacturing activities and the production resources. The available production resources and inhouse capacity must be adequate, which deserves to enhance the current resources utilization to fulfil the critical tasks or customer demands (Baker, 1974; Leitão, 2009). The production environment also contains high levels of uncertainty, which leads to a more complex situation.

Flowshop scheduling as a class of scheduling problems is a particular case of production scheduling. Practitioners use different techniques to solve industrial scheduling problems, and recently, most of the studies have turned to deal with simulation and artificial intelligence techniques. These methods have demonstrated a significant step forward in solving Flowshop Scheduling (FSS) problem with less computational exertion and more powerful results. Given this section, a brief survey of the FSS issues is presented based on previous works (Pinedo, 2012; Ruiz & Maroto, 2005).

2.3 Flowshop Scheduling

A sound scheduling system will contribute directly to the performance of a manufacturing enterprise. In industries, the most common scheduling layout is the flowshop scheduling problem (FSP). Additionally, the practitioners apply the FSP in a manufacturing environment where a family of products (with some similar features) to be manufactured (Fan & Winley, 2008; Moccellin, 1995; Naderi & Ruiz, 2014a)

Flowshop scheduling is one of three main problems of shop scheduling; the other two problems are open shop scheduling and the job shop. Generally, if the job shop has *n* jobs to be processed through *m* machine. In this case, the number of possible sequences is (n!) m, where *n* represents the number of jobs, and *m* represents the number of machines, as shown in Figure 2.1. The flowshop scheduling problem is consequent of the job shop where all jobs contain the same technological sequence. It has the same amount of operations for each route of machines, and each job should pass the same. The objective is to find the best sequence of jobs based on applying for the same order according to technological constraints. The decision of scheduling problems will concern two bases (Jemni & Ladhari, 2011; Park, 1976):

i. At least two machines should consider the sequence for orders of the jobs.
ii. The machine loading schedule considers the time sequence of each machine for different jobs.

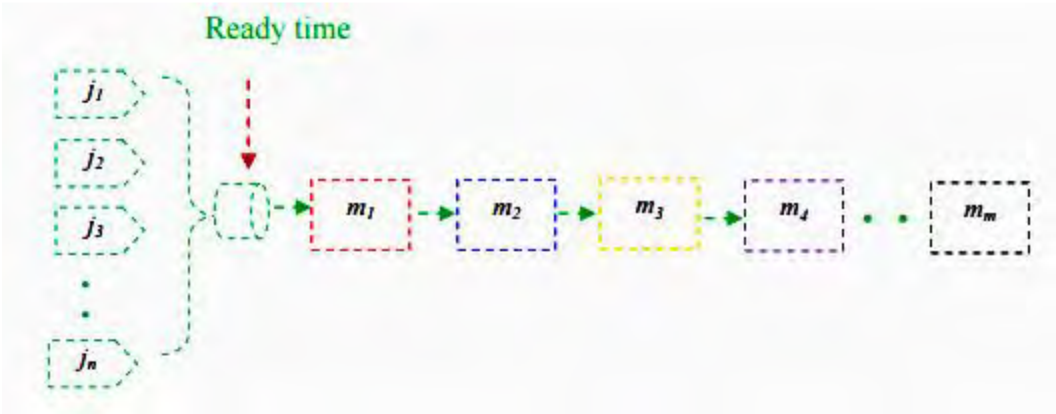

Figure 2.1: Flowshop problem (Garey, Johnson, & Sethi, 2008)

For that, researchers agree to consider the flowshop scheduling problem is NP-hard problem (Garey, Johnson, & Sethi, 2008b; Park, 1976). Furthermore, the first step to investigate the effective optimization method in the scheduling problem is by specifying the scheduling objectives. The objective classification paves the way to analyses the critical parameters to detect the root cause.

2.3.1 NP-Hard Problems

NP-hard (non-deterministic polynomial-time hard) is a class of the hardest problems in computer science. Non-deterministic means that to get the solution of a problem, one has to go through the trial and error process (Laudis, Shyam, Suresh, & Kumar, 2018). The is no exact solution for the NP-hard problems but it gets better after the iterations process. The obtained solution from the optimization process will always converge near to the exact solution (Mahapatra, Dash, & Pradhan, 2017).

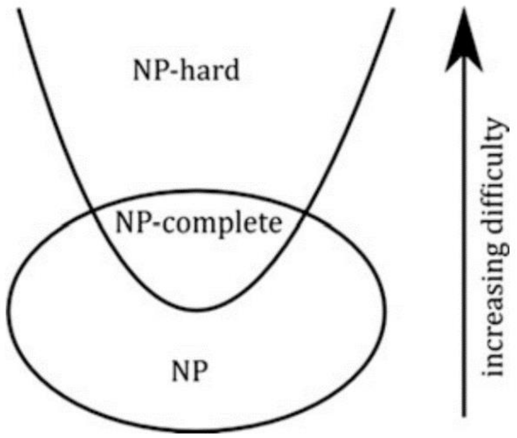

Figure 2.2: Hierarchy of problems based on solving difficulties (Laudis et al., 2018)

NP-hard is a class of problems that have the same or higher level of hardness compared to the hardest problems in NP (refer to Figure 2.2). Generally, P-class are problems that can be solved in polynomial time and NP-class problems are harder than the P-class. The problems under this category are hard to solve but easy to check (Mahapatra et al., 2017). NP-problems can be solved by the non-deterministic Turing machine in polynomial time (Laudis et al., 2018; Woeginger, 2003a).

NP-hard is a sophisticated way to describe the complex and difficult problems and it is also considered as the initial steps to the development of approximate algorithms which are not able to provide an exact answer for a problem because these algorithms are only capable of finding a near-optimum solution (Woeginger, 2003b; Yang, 2008).

The NP-hard problems are common and available in various fields such as approximate computing, configuration, cryptography, data mining, decision support, planning, process monitoring and control, routing/vehicle routing, scheduling and phylogenetics (Yang, 2008).

Product manufacturing may involve a few jobs that are completed by several machines. Each job has to be performed sequentially on different machines in the production line. This kind of scenario is known as Flowshop. Additionally, the process focus on doing the operations in the flowshop on all jobs in the same order (sequence). All jobs need to go through the same series of machines in a predetermined sequence, and the machines are in a series arrangement. Usually, the problem to be solved in a flowshop is to find the minimum completion time of the processing time of the last job (on the last machine). This kind of scenario is also known as the makespan (Jemni & Ladhari, 2011; Pinedo, 2012; Eric Taillard, 1990).

To depict an example of flowshop scheduling, Table 2.1 shows the lists of processing times for each job on a particular machine. The makespan value is equal to the total time needed to complete all jobs, and based on the example in Table 2.1, the makespan value is 47 minutes (refer Figure 2.3).

Table 2.1: Processing time example of 6 jobs in 3 machines

Number of Jobs	**Number of Machines**		
	Machine 1	**Machine 2**	**Machine 3**
Job 1	5	6	8
Job 2	4	5	7
Job 3	7	5	6
Job 4	6	4	5
Job 5	5	3	6
Job 6	8	3	4

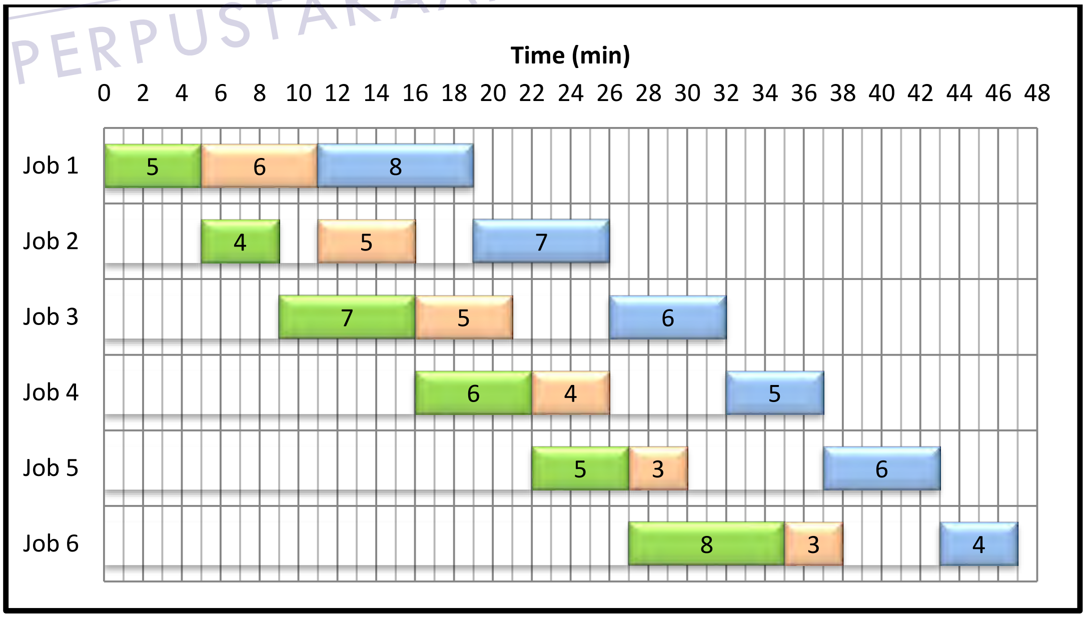

Figure 2.3: Gantt chart for the flowshop. The green, orange, and blue coloured bars represent Machine 1, Machine 2 and Machine 3 respectively (The number in each bar is the processing time in minutes) (Sidek & Bareduan, 2014)

According to Figure 2.3, the waiting time is the gap between jobs. In Permutation Flowshop Scheduling Problem (PFSP), it is imperative to minimise the waiting time as much as possible. From here, an effective arrangement of jobs (permutation) is the way to reduce the amount of waiting times.

Managing a production process that has some involvement with the scheduling process is not an easy task. Moreover, the literature manages to prove that the flowshop scheduling problem is under the NP-hard category (Garey et al., 2008b; Park, 1976). Researchers proposed some heuristics over the years such as Palmer heuristic (Palmer, 1965), Gupta heuristic (Gupta, 1971), CDS heuristic Campbell et al. (1970), and NEH (Nawaz, Enscore, and Ham, 1983b) heuristic. NEH heuristics is the leading heuristic in the field of flowshop scheduling. However, this heuristic also has its shortcoming and research conducted to overcome the shortcoming of NEH (Fan & Winley, 2008).

The heuristic approaches to solve the flowshop scheduling problem now developed with the adaptation of particle swarm optimization on it. For instance for flowshop scheduling problems where the researcher is finding the makespan and lateness minimization using the Particle Swarm Optimization algorithm (Tasgetiren et al., 2007a)

Researchers also used the ant colony algorithm to solve the permutational flowshop scheduling problems, which are to minimize the makespan and outsourcing cost in a single machine environment (Tavares Neto & Godinho Filho, 2011). Apart from the particle swarm optimization and ant colony, some studies used the artificial bee colony algorithm to solve the flowshop scheduling problem (Han, Duan, & Zhang, 2011; Liu & Liu, 2013; Pan, Tasgetiren, Suganthan, & Chua, 2011; Tasgetiren et al., 2010).

2.3.2 Scheduling Objectives

Scheduling systems using the heuristic approaches facilitate the matching activity decisions of a set of tasks within a period (Bard, 1995). Many researchers concern with the real problems and in the last two decades, a significant amount of research activities focused on the manufacturing management. Cerdá, Henning, & Grossmann (1997) discussed the multi-objective problem proposition intending to minimize total tardiness, makespan, and the total number of tardy orders. The researchers used three separate objective functions. Hui & Gupta (2001) present a method to handle two separate objective functions. The presented method aimed to minimize makespan and tardiness. Chen, Lu, & Yu (2002) extend the objectives to five functions for the same type of problem.

Loukil, Teghem, & Tuyttens (2005) present a multi-objective simulated annealing method. The method concern with the multi-objective production scheduling able to design a general method based on efficient schedules for a broad set of scheduling models. The researchers introduce the models to treat one machine, parallel machines, and permutation flowshops and the corresponding notations.

Janak, Floudas, Kallrath, & Vormbrock (2006) introduce a mathematical production scheduling model for multi-purpose batch plants based on multi-product industries. The objective function concerns the weighted individual functions. In this method, the main target is to maximize sales in parallel with the minimizing number of binary variables, orders satisfaction, overall demand satisfaction, resources, orders, due date order, and inventory level. Quadt & Kuhn (2007) introduce a taxonomy for flow lines with parallel machines scheduling procedures. They divide the problem concerning the production stages, the individual jobs, or the sub-problems to be solved (batching, loading, and sequencing).

Paternina-Arboleda, Montoya-Torres, Acero-Dominguez, & Herrera-Hernandez (2008) propose a k -stage Jobs Schedule algorithm based on the identification and exploitation of the bottleneck stage. They consider minimizing the makespan problem of k stages and ms machines at any stage. The results observed smaller variance and less computational requirements.

Kulcsar & Forrai (2009) propose new modelling based on rescheduling the discrete production scheduling. They solve the problem of multi-objective scheduling and rescheduling using a new interpretation relational operator. The results were encouraging for the application of the method in multi-objective optimization problems. Based on the present literature, the author can conclude that the scheduling objectives can be classified into four types as presented in the next sections.

2.3.2.1 Minimize the Flow Time

The problem of minimizing average flow time on identical parallel machines has received much attention in the past few years. Leonardi & Raz (2007) showed that the Shortest Remaining Processing Time (SRPT) algorithm has a competitive ratio. The same authors also showed a matching lower bound on the competitive ratio of any on-line (randomized) algorithm for this problem. It is worth to note that this setting produced the leading results in the off-line setting of this problem. Garg, Kumar, & Muralidhara (2008) gave on-line algorithms for minimizing average flow-time on related machines with a poly-logarithmic competitive ratio. Much less is known when we impose the constraint that a job can be scheduled only on a subset of machines.

2.3.2.2 Minimize the Makespan

Minimize the makespan means to complete all jobs as soon as possible. Several published papers focused on the flowshop problem for makespan minimization. For instance, Grabowski & Pempera (2007) propose two tabu search (TS) algorithms. Wang, Zhang and Zheng (2006) proposed a hybrid genetic algorithm (HGA), Liu, Wang and Jin (2008) proposed a hybrid algorithm based on particle swarm optimization (HPSO) and Qian, Wang, Huang, Wang and Wang (2009) took the differential evolution (DE) as inspiration and later, Qian, Wang, Huang and Wang (2009) adapted the methodology to the multicriteria case. Wang, Pan, Suganthan, Wang, & Wang, (2010) proposed a Hybrid Discrete Differential Evolution (HDDE) and Ribas, Companys, & Tort-Martorell (2011) proposed an iterated greedy (IG) algorithm to work on the makespan minimization.

2.3.2.3 Minimize Job Lateness

One of the more common scheduling problems in batch production involves the trade-off between achieving batch size efficiency and meeting customer due dates. At one extreme, scheduling large batches means that relatively little time spent for setup, and, as a result, efficiency is high. However, long runs on a given product may mean that the company will miss the due dates for other products. At the other extreme, scheduling based on due dates aligns priorities with customer needs, but the shorter runs mean more substantial amounts of setup time, and, as a result, capacity may become inadequate to meet demand on time (McMahon & Florian, 1975; Uzsoy, Lee, & Martin-Vega, 1992; Hinder & Mason, 2017; Allahverdi & Allahverdi, 2018).

2.3.2.4 Minimize Average Tardiness

Average tardiness represents the order time difference due to the actual completion time and the due date. Scheduling, according to this performance measure, helps companies offer a high service level to their customers, which is essential for survival in the market. To minimize job earliness and tardiness, Zhu & Heady (2000) developed a mixed-integer programming formulation for multi-machine scheduling problems. The study managed to come out with a model that is useful for finding the optimal solutions for problems with nine jobs and three machines. Yalaoui & Chu (2002) used an exact method to minimize the tardiness in an identical parallel machine scheduling. The study proposed a branch and bound (BAB) algorithm to do the optimization which covers the theoretical properties, upper and lower bounds.

2.4 Metaheuristic Optimization methods

Practitioners use scheduling as a decision-making practice regularly in many manufacturing and services industries. For this reason, researchers developed several algorithms to optimize one or more objectives with the allocation of resources to work on over given periods. In this section, the researcher will provide a survey of the most prominent and successful meta-heuristic algorithms used to solve the scheduling problem. Heuristic techniques are any method for solving hard problems. The researchers used these methods to speed up the process of finding a satisfactory solution when classic methods are less efficient or for finding an approximate solution when classic methods fail to find any exact solution.

The algorithms managed to achieve this excellent feature by trading optimality, completeness, accuracy, or precision for speed. In a way, it can be considered a shortcut. Principal considerations in creative problem solving are the need for an adequate knowledge base. The type of heuristics that the practitioners choose to apply to a given problem area is "vitally important but largely unexplored". Thus, the method maintains the feasible solution throughout the procedure of searching proceed by moving from one feasible solution to one of its neighbours while improving the objective function (Gupta, Zanakis, & Mandokovic, 1988).

Johnson (1954) proposed a method of scheduling for two machines with the aim to find the optimal sequence for the jobs to minimize makespan. The method managed to reduce the amount of idle time between the two machines and it is also worked with three machines but with some restrictions. Lee (1997) provided the extension of the Johnson's rule (method) by considering the constraint in the availability of machines. This is because in some scenarios, one of the machines need to be stopped due to breakdown or preventive maintenance and the production still must proceed. From here, Lee (1997) proposed two-time heuristic algorithms.

Palmer algorithm is used to perform scheduling in a simple flowshop with more than two machines and the objective of the algorithm is to achieve the minimum completion time. One advantage of using Palmer algorithm is the low computational time complexity because of the simplicity of the algorithm (Palmer, 1965). Hong, Huang and Horng (2006) proposed a combination of Longest-Processing-Time-first (LPT) and Palmer algorithms to solve flexible flow-shop problems with more than two machines. The propose algorithm generated competitive results and with more time allocated for the computation, better results can be achieved. Hundal and Rajgopal (1988) proposed an extension of Palmer heuristics by adding some set of computational mechanism. With the minor extension, the heuristic managed to perform better than other more sophisticated competitor.

In the field of scheduling, several rules are commonly used by the researchers and industry partners. One of the common rules is the First Come First Serve (FCFS) which is also known as First In First Out (FIFO). In this rule, the job is processed as soon as it enters the production system (Xoxa, Zotaj, Tafa & Fejzaj, 2014). Another rule is the Last Come, First Served (LCFS) where the last job that enter the queue will be processed first. An example for LCFS is in a warehouse where the items are stacked on each other and when they need to be used, the operator will choose the uppermost item first (Jouini, 2012). He and Alfa (1998) reported that in the field of advanced telephone networks, LCFS is more efficient than FCFS.

The Earliest Due Date (EDD) rule prioritizes jobs based on the due dates where the job with the nearest due date will be processed first. EDD scheduling is ideal for products that are not behind-schedule in the context of many problem formulations (Wu, Yin & Cheng, 2011). Roychowdhury, Allen and Allen (2017) proposed a GA based on the EDD rule for the automotive stamping operations. Another well-known rule is the Shortest Processing Time (SPT) which arrange jobs in the order of increasing processing times. The job with the shortest processing time will be worked on first and this rule is able to minimize the number of jobs completed at any point (Zhou, Feng & Han, 2001). According to Qi, Bard and Yu (2006), although SPT can minimize the average flow time and average number of jobs, it will be pushed back the long job in the schedule.

In hard problems and to find efficient optimal solutions, meta-heuristics algorithms are higher-level heuristic that designed to find, generate, or select a heuristic (partial search algorithm). Meta-heuristics algorithms are the iterative generation process that guides a secondary heuristic for exploring and exploiting the search space. The literature divides the meta-heuristics algorithms into two categories; trajectory and population-based searches (Blum & Roli, 2003; Talbi, 2009; Toutouh, 2015), as shown in Figure 2.4.

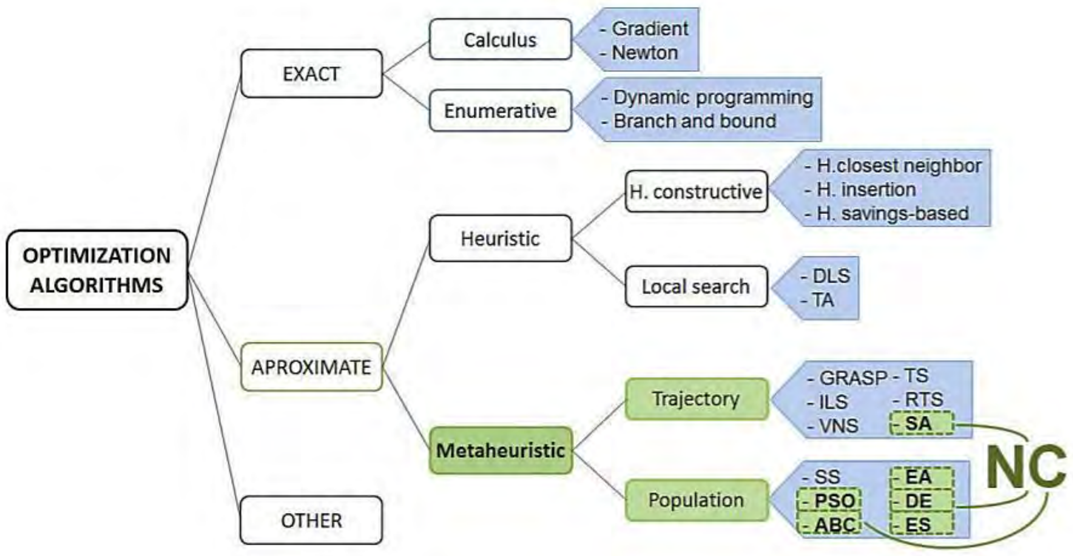

Figure 2.4: General classification of the optimization techniques (Toutouh, 2015)

Natural Computing (NC) is a process of developing artificial systems inspired by nature. The inspiration from nature is used to develop algorithms to solve complex problems. The first mathematical model of a neuron, proposed by McCulloch & Pitts (1943) becomes the foundation base of Artificial Neural Network. After that, nature-inspired optimization techniques are divided into three groups (de Castro, 2006; Toutouh, 2015):

i. Evolutionary Computing (EC), that utilize the ideas of the evolution of species to develop Evolutionary Algorithms (EAs).
ii. Swarm Intelligence (SI), that is inspired by the movement and behaviour of organisms.
iii. Artificial Immune Systems (AIS), that is used to develop models based on the immune systems.

However, there is also a bio-inspired algorithm that does not fall into any categories mentioned above such as Simulated Annealing (SA).

2.4.1 PSO Optimization

The PSO algorithm has several features, such as a new encoding scheme, the implementation of the best velocity equation and neighbourhood topology among several different variants, and effective incorporation of local search. For this reason, researchers applied this algorithm to job schedule processes (Yi, 2016). PSO is known to exploits particles to search for areas with high-quality solutions. Each particle moves in the search space in the preset velocity with the previous high-quality memories as guidance (Md Fauadi & Murata, 2010).

Tasgetiren, Sevkli, Liang, & Gencyilmaz (2004) proposed a discrete particle swarm optimization algorithm to determine a sequence of n jobs for the method to process through *m* machines that minimize the number of tardy jobs. The group implemented the algorithms using the due date configurations of the data sets by Demirkol, Mehta, & Uzsoy (1998). The study concluded that the particle swarm optimization algorithm gives promising solutions utilizing the proposed SPV (Smallest Position Value) heuristic rule.

Liao, Tseng, & Luarn (2007) presented a PSO-LS algorithm by extending from discrete PSO for flowshop scheduling. They developed the particle movement to the new sequence and incorporated a local search scheme into the proposed algorithm. Computational results show that the proposed algorithm is exceptionally modest in case of a total flowtime criterion.

Lian, Gu, and Jiao (2008) presented a novel particle swarm optimization (NPSO) algorithm, which applied to permutation flow-shop scheduling to minimize makespan. The algorithm applied based on the discrete characteristic of FSSP, and the results observed more efficacious than standard GA for FSSP to minimize makespan.

Tseng and Liao (2008) solved the multistage hybrid flow-shop scheduling problem with multiprocessor tasks by using particle swarm optimization (PSO). Based on the results, the proposed PSO algorithm seems to perform better compared to all the existing algorithms for the considered problem.

Sha and Hung Lin (2009) proposed a particle swarm optimization-based multiobjective algorithm for flowshop scheduling to meet the requirements of realistic manufacturing systems. The proposed evolutionary algorithm searches the Pareto optimal solution for objectives by considering the makespan, mean flow time, and machine idle time. The results show that the modified particle swarm optimization algorithm performed better in terms of searching quality and efficiency than other traditional heuristics.

Li, Pan, and Mao (2014) presented a hybrid algorithm by combining particle swarm optimization (PSO) and iterated local search (ILS) for solving the hybrid flowshop scheduling (HFS) problem with preventive maintenance (PM) activities. The ILS-based local search procedure is embedded in the algorithm to improve the exploitation ability of the proposed algorithm. Detailed comparisons verify the efficiency and effectiveness of the proposed algorithm.

Ramanan, Iqbal, and Umarali (2014) used a particle swarm optimization (PSO) approach for optimizing the makespan of an FSSP. In this method, the group employed a Variable neighbourhood search (VNS) to overcome the early convergence of the PSO and helps in global search. The experimental results show that the solution quality of FSSP can be improved if the method can direct the search in a quality space based on the proposed PSO approach (PSO-NEH-VNS).

Given the swarm behaviour is tested for solving a combinatorial optimization problem such as a sequencing problem under constraints. The computational results show that this approach outperforms the compared methods in terms of the quality of solutions in short time requirements. Also, the researchers evaluated the performance of the proposed approach according to a real-world industrial problem.

The reasons for PSO algorithm to be favourable by some researchers because of its simplicity and easy implementation. However, PSO algorithm tends to fall easily into local optimum especially in complex problems (Choi, Ohmori, Yoshimoto, & Ohtake, 2010). The algorithm also suffers the issue of premature convergence and this is the main reason for the improvements made by previous researchers (Abdmouleh et al., 2017; Li, Du, & Nian, 2014; Yi, 2016). PSO also known to be problematic with a problems of scattering and the task to define the initial design parameters might not be straight forward (Abdmouleh et al., 2017).

2.4.2 Ant Colony Optimization

ACO algorithms are population-based search algorithms based on the food hunting patterns of real ants that utilize agents (ants), single or multiple, to construct the optimal solution iteratively.

Rajendran and Ziegler (2004) focused on the problem of scheduling in permutation flowshops by using two ACO algorithms to minimize the sum of the total flow time of jobs and makespan. The group evaluated the effectiveness of the proposed ant-colony algorithms by considering the benchmark problems and upper bound values for the makespan given by E. Taillard (1993). It has found that both the proposed ant-colony algorithms perform well than the existing ant-colony algorithm in the case of relatively large-sized than small-sized permutation flowshop problems.

Shyu, Lin, and Yin (2004) developed an ACO-based algorithm to solve the two-machine flowshop scheduling problem with no waiting between operations and including set up time. They have shown that the ACO algorithms outperform previous algorithms and considered to be effective and robust in dealing with the said scheduling problems.

Rajendran and Ziegler (2005) proposed two ACO-based algorithms to minimize the total flow time in permutation flow-shops. The algorithm generates an initial seed sequence and carries out a local search at the end of each iteration. The authors observed that it is not possible to identify one heuristic that is best for the entire set of benchmark problems.

Yagmahan and Yenisey (2008) introduced ACO for minimizing multi-objectives, including makespan, total flow time, and total machine idle time. The computational results show that the proposed algorithm is more effective and better than other methods compared.

Zhou, Lee, and Nee (2008) proposed ant colony optimization for dynamic jobs scheduling problems and showed that ACO performed excellently, but the however performance of ACO does not improve with increasing the iterations and ants per iteration. They have also concluded that the ACO can be enhanced when the machine utilization increases.

Yagmahan and Yenisey (2010) considered the flowshop scheduling problem concerning the minimization of both objectives of makespan and total flow time. The proposed algorithm based on the ACS metaheuristic called a multi-objective ant colony system algorithm (MOACSA). They concluded that the proposed MOACSA performs better than CR (MC) algorithm, HAMC algorithms, and GA for said multi-objective flowshop scheduling problem.

Rabanimotlagh (2011) developed the ant colony optimization algorithm to solve the permutation flowshop scheduling problem. This problem is optimized considering two criteria, makespan and total flow time. Then the results observed that the proposed approach performs best among all other algorithms.

Zhong and Zhang (2012) improved ACO algorithm to solve the PFSP, which takes the minimum of makespan as the objective function. Also, they integrate NEH heuristic with ACO for scheduling problems cooperatively, define the heuristic information of ACO via makespan increment, and come up with a new priority rule for PFSP. The experiment results show that the proposed algorithm is productive and competitive.

Chen, Zhang, and Ma (2013) present a novel hybrid ant colony optimization (ACO&VNS) to solve the permutation flow-shop scheduling problem (PFS) in manufacturing systems and industrial processes. The main feature of this hybrid algorithm is to hybridize the solution construction mechanism of the ACO with variable neighbourhood search (VNS), which can also be embedded into the ACO algorithm as a neighbourhood search to improve solutions. Moreover, the hybrid algorithm considers both solution diversification and solution quality. The experimental results for benchmark PFS instances have shown that the hybrid algorithm is very efficient in solving the permutation flow-shop scheduling in manufacturing engineering compared with the best existing methods in terms of solution quality.

ACO is suitable for problems with specific and predefined sources and destinations. It is an algorithm with great capabilities, but its weaknesses should be taken into consideration. The theoretical analysis of utilizing ACO is not easy and it requires some extra effort with the initial settings. ACO is also known to be not independent because it operates on the sequence of random decisions (Selvi & Umarani, 2010). The probability distribution of ACO are said to change by iteration and it also suffers the weakness of poor global search ability (Cui & Han, 2013; Zhang, Xiao, & Fei, 2017). The convergence is guaranteed for ACO but the time for convergence is uncertain. Some researchers mentioned the convergence rate of the

REFERENCES

Abdmouleh, Z., Gastli, A., Ben-Brahim, L., Haouari, M., & Al-Emadi, N. A. (2017). Review of optimization techniques applied for the integration of distributed generation from renewable energy sources. *Renewable Energy*, *113*(2017), 266–280. https://doi.org/10.1016/j.renene.2017.05.087

Abdollahpour, S., & Rezaeian, J. (2015). Minimizing makespan for flow shop scheduling problem with intermediate buffers by using hybrid approach of artificial immune system. *Applied Soft Computing Journal*, *28*, 44–56. https://doi.org/10.1016/j.asoc.2014.11.022

Ahmadizar, F. (2012). A new ant colony algorithm for makespan minimization in permutation flow shops. *Computers & Industrial Engineering*, *63*(2), 355–361. https://doi.org/10.1016/J.CIE.2012.03.015

Akay, B., & Karaboga, D. (2012). Artificial bee colony algorithm for large-scale problems and engineering design optimization. *Journal of Intelligent Manufacturing*, *23*(4), 1001–1014. https://doi.org/10.1007/s10845-010-0393-4

Alzaqebah, M., & Abdullah, S. (2011). Artificial bee colony search algorithm for examination timetabling problems. *International Journal of the Physical Sciences*, *6*(17), 4264–4272. https://doi.org/10.5897/IJPS11.200

Anam, S. (2017). Multimodal optimization by using hybrid of artificial bee colony algorithm and BFGS algorithm. *Journal of Physics: Conference Series*, *893*(1). https://doi.org/10.1088/1742-6596/893/1/012068

Arnold, S. F. (2006). Design of Experiments with MINITAB. *The American Statistician*, *60*(2), 205–205. https://doi.org/10.1198/tas.2006.s46

Ayan, K., & Kiliç, U. (2012). Artificial bee colony algorithm solution for optimal reactive power flow. *Applied Soft Computing Journal*, *12*(5), 1477–1482. https://doi.org/10.1016/j.asoc.2012.01.006

Baker, K. R. (1974). *Introduction to sequencing and scheduling*. John Wiley & Sons.

Balaji, A. N., & Porselvi, S. (2014). Artificial immune system algorithm and simulated annealing algorithm for scheduling batches of parts based on job availability model in a multi-cell flexible manufacturing system. *Procedia Engineering*, *97*, 1524–1533. https://doi.org/10.1016/j.proeng.2014.12.436

Baldwin, A., & Bordoli, D. (2014). Handbook for Construction Planning and Scheduling. In *Handbook for Construction Planning and Scheduling*. https://doi.org/10.1002/9781118838167

Ballard, G. (2000). Phase scheduling. *LCI White Paper*, *7*.

Bandyopadhyay, S. (2019). Production and Operations Analysis. In *Production and Operations Analysis*. *149*(2020). https://doi.org/10.1201/9781351113670

Banharnsakun, A., Achalakul, T., & Sirinaovakul, B. (2011). The best-so-far selection in Artificial Bee Colony algorithm. *Applied Soft Computing Journal*, *11*(2), 2888–2901. https://doi.org/10.1016/j.asoc.2010.11.025

Banharnsakun, A., Sirinaovakul, B., & Achalakul, T. (2012). Job shop scheduling with the Best-so-far ABC. *Engineering Applications of Artificial Intelligence*, *25*(3), 583–593. https://doi.org/10.1016/j.engappai.2011.08.003

Bard, J. F. (1995). Heuristic Scheduling Systems with Applicatians to Production Systems and Project Management [Book Re. *IEEE Transactions on Engineering Management*, *42*(4), 422. https://doi.org/10.1109/tem.1995.482094

Benbouzid-Si Tayeb, F., & Belkaaloul, W. (2014). Towards an artificial immune system for scheduling jobs and preventive maintenance operations in flowshop problems. *IEEE International Symposium on Industrial Electronics*, 1065–1070. https://doi.org/10.1109/ISIE.2014.6864761

Bernardino, H. S., Barbosa, H. J. C., Lemonge, A. C. C., & Fonseca, L. G. (2008). A new hybrid AIS-GA for constrained optimization problems in mechanical engineering. *2008 IEEE Congress on Evolutionary Computation, CEC 2008*, 1455–1462. https://doi.org/10.1109/CEC.2008.4630985

Bin Md Fauadi, M. H. F., & Murata, T. (2010). Makespan minimization of machines and Automated Guided Vehicles schedule using Binary Particle Swarm Optimization. *Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010*, 1897–1902. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.302.841

Blum, C., & Roli, A. (2003). Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. *ACM Computing Surveys*, Vol. 35, pp. 268–308. https://doi.org/10.1145/937503.937505

Campbell, H. G., Dudek, R. A., & Smith, M. L. (1970). A heuristic algorithm for the n job, m machine sequencing problem. *Management Science*, *16*(10), B-630.

Carlier, J & Chretienne, P. (1988). *Problèmes d'ordonnancement*, Masson, Paris: Col. ERI.

Cerdá, J., Henning, G. P., & Grossmann, I. E. (1997). A Mixed-Integer Linear Programming Model for Short-Term Scheduling of Single-Stage Multiproduct Batch Plants with Parallel Lines. *Industrial and Engineering Chemistry Research*, *36*(5), 1695–1707. https://doi.org/10.1021/ie9605490

Chakaravarthy, G. V., Marimuthu, S., Ponnambalam, S. G., & Kanagaraj, G. (2014). Improved sheep flock heredity algorithm and artificial bee colony algorithm for scheduling m-machine flow shops lot streaming with equal size sub-lot problems. *International Journal of Production Research*, *52*(5), 1509–1527.

Chen, C. F., Daud, N. Bin, Zhou, K. Q., & Elisha Nyamasvisva, T. (2019). A Modified Artificial Bee Colony Algorithm for Function Optimization Strategy. *IOP Conference Series: Materials Science and Engineering*, *551*(1). https://doi.org/10.1088/1757-899X/551/1/012029

Chen, J., Zhang, X. X., & Ma, Y. Y. (2013). A hybrid ant colony optimization algorithm for permutation flow-shop scheduling in manufacturing systems and industrial process. *Applied Mechanics and Materials*, *345*, 438–441. https://doi.org/10.4028/www.scientific.net/AMM.345.438

Chen, M. H., Chang, P. C., & Lin, C. H. (2014). A self-evolving artificial immune system II with T-cell and B-cell for permutation flow-shop problem. *Journal of Intelligent Manufacturing*, *25*(6), 1257–1270. https://doi.org/10.1007/s10845-012-0728-4

Chen, R. S., Lu, K. Y., & Yu, S. C. (2002). A hybrid genetic algorithm approach on multi-objective of assembly planning problem. *Engineering Applications of Artificial Intelligence*, *15*(5), 447–457. https://doi.org/10.1016/S0952-1976(02)00073-8

Choi, H., Ohmori, S., Yoshimoto, K., & Ohtake, H. (2010). Improvement of Particle Swarm Optimization: Application of the mutation concept for the escape from local minima. *2010 8th International Conference on Supply Chain Management and Information*, 1–5.

Chong, C. S., Low, M. Y. H., Sivakumar, A. I., & Gay, K. L. (2007). Using a bee colony algorithm for neighborhood search in job shop scheduling problems. *21st European Conference on Modelling and Simulation: Simulations in United Europe, ECMS 2007*, 459–465. https://doi.org/10.7148/2007-0459

Chong, C. S., Sivakumar, A. I., Low, M. Y. H., & Gay, K. L. (2006a). A bee colony optimization algorithm to job shop scheduling. *Proceedings - Winter Simulation Conference*, 1954–1961. https://doi.org/10.1109/WSC.2006.322980

Chong, C. S., Sivakumar, A. I., Low, M. Y. H., & Gay, K. L. (2006b). A bee colony optimization algorithm to job shop scheduling. *Proceedings - Winter Simulation Conference*, 1954–1961. https://doi.org/10.1109/WSC.2006.322980

Choudhury, B. B., Mishra, D., & Biswal, B. B. (2008). Task assignment and scheduling in a constrained manufacturing system using GA. *International Journal of Agile Systems and Management*, *3*(1–2), 127–146. https://doi.org/10.1504/IJASM.2008.019603

Cohen, J. (2013). Statistical Power Analysis for the Behavioral Sciences. In *Statistical Power Analysis for the Behavioral Sciences*. https://doi.org/10.4324/9780203771587

Cui, S., & Han, S. (2013). Ant colony algorithm and its application in solving the traveling salesman problem. *Proceedings - 3rd International Conference on Instrumentation and Measurement, Computer, Communication and Control, IMCCC 2013*, 1200–1203. https://doi.org/10.1109/IMCCC.2013.266

Dahiya, S. S., Chhabra, J. K., & Kumar, S. (2010). Application of Artificial Bee Colony Algorithm to Software Testing. *2010 21st Australian Software Engineering Conference*, 149–154. https://doi.org/10.1109/ASWEC.2010.30

de Castro, L. N., & Timmis, J. I. (2003). Artificial immune systems as a novel soft computing paradigm. *Soft Computing*, *7*(8), 526–544. https://doi.org/10.1007/s00500-002-0237-z

de Castro, Leandro Nunes. (2006). Fundamentals of Natural Computing. In *Fundamentals of Natural Computing*. https://doi.org/10.1201/9781420011449

Demirkol, E., Mehta, S., & Uzsoy, R. (1998). Benchmarks for shop scheduling problems. *European Journal of Operational Research*, *109*(1), 137–141. https://doi.org/10.1016/S0377-2217(97)00019-2

Deng, G., Xu, Z., & Gu, X. (2012). A discrete artificial bee colony algorithm for minimizing the total flow time in the blocking flow shop scheduling. *Chinese Journal of Chemical Engineering*, *20*(6), 1067–1073.

https://doi.org/10.1016/S1004-9541(12)60588-6

Dong, Y., Liu, H. J., Xu, B. D., & Tang, J. F. (2005). Modified genetic algorithm based on real-coding. *Dongbei Daxue Xuebao/Journal of Northeastern University*, *26*(4), 321–323. Retrieved from http://en.cnki.com.cn/Article_en/CJFDTotal-DBDX200504005.htm

Doyen, A., & Engin, O. (2007). A New Approach to Solve Flowshop Scheduling Problems By Artificial Immune Systems. *Doğuş Üniversitesi Dergisi*, *1*(8), 12–27. https://doi.org/10.31671/dogus.2019.238

Emmons, H., & Vairaktarakis, G. (2013). *Flow Shop Scheduling Theoretical Results, Algorithms, and Applications*. New York: Springer.

Engin, O., & Döyen, A. (2004). A new approach to solve hybrid flow shop scheduling problems by artificial immune system. *Future Generation Computer Systems*, *20*(6 SPEC. ISS.), 1083–1095. https://doi.org/10.1016/j.future.2004.03.014

Erenay, F. S., Sabuncuoglu, I., Toptal, A., & Tiwari, M. K. (2010). New solution methods for single machine bicriteria scheduling problem: Minimization of average flowtime and number of tardy jobs. *European Journal of Operational Research*, *201*(1), 89–98. https://doi.org/10.1016/j.ejor.2009.02.014

Ettefagh, M. M., & Javash, M. S. (2014). Optimal synthesis of four-bar steering mechanism using AIS and genetic algorithms. *Journal of Mechanical Science and Technology*, *28*(6), 2351–2362. https://doi.org/10.1007/s12206-014-0526-3

Fan, J. P.-O., & Winley, G. K. (2008). A heuristic search algorithm for flow-shop scheduling. *Informatica*, *32*(4). 453-464.

Fernandez-Viagas, V., Perez-Gonzalez, P., & Framinan, J. M. (2018). The distributed permutation flow shop to minimise the total flowtime. *Computers & Industrial Engineering, 118*, 464-477.

Fernandez-Viagas, V., & Framinan, J. M. (2015). NEH-based heuristics for the permutation flowshop scheduling problem to minimise total tardiness. *Computers & Operations Research*, *60*, 27–36. https://doi.org/http://dx.doi.org/10.1016/j.cor.2015.02.002

Fernandez-Viagas, V., Dios, M., & Framinan, J. M. (2016). Efficient constructive and composite heuristics for the permutation flowshop to minimise total earliness and tardiness. *Computers & Operations Research, 75*, 38-48.

Framinan, J. M., Leisten, R., & Rajendran, C. (2003). Different initial sequences for the heuristic of Nawaz, Enscore and Ham to minimize makespan, idletime or flowtime in the static permutation flowshop sequencing problem. *International Journal of Production Research*, *41*(1), 121–148. https://doi.org/10.1080/00207540210161650

Gaafar, L. K., Masoud, S. A., & Nassef, A. O. (2008). A particle swarm-based genetic algorithm for scheduling in an agile environment. *Computers and Industrial Engineering*, *55*(3), 707–720. https://doi.org/10.1016/j.cie.2008.01.020

Gao, J., & Chen, R. (2011). A hybrid genetic algorithm for the distributed permutation flowshop scheduling problem. *International Journal of Computational Intelligence Systems*, *4*(4), 497–508. https://doi.org/10.1080/18756891.2011.9727808

Gao, J., Chen, R., & Deng, W. (2013). An efficient tabu search algorithm for the distributed permutation flowshop scheduling problem. *International Journal of*

Production Research, *51*(3), 641–651. https://doi.org/10.1080/00207543.2011.644819

Gao, S., & Cao, C. (2012). Convergence analysis of particle swarm optimization algorithm. *Advances in Information Sciences and Service Sciences*, *4*(14), 25–32. https://doi.org/10.4156/AISS.vol4.issue14.4

Garey, M. R., Johnson, D. S., & Sethi, R. (2008). The Complexity of Flowshop and Jobshop Scheduling. *Mathematics of Operations Research*, *1*(2), 117–129. https://doi.org/10.1287/moor.1.2.117

Garg, N., Kumar, A., & Muralidhara, V. N. (2008). Minimizing total flow-time: The unrelated case. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *5369 LNCS*, 424–435. https://doi.org/10.1007/978-3-540-92182-0_39

Goldberg, D. (1989). Genetic algorithms in search, optimization, and machine learning. *Choice Reviews Online*, *27*(02), 27-0936-27–0936. https://doi.org/10.5860/choice.27-0936

Grabowski, J., & Pempera, J. (2007). The permutation flow shop problem with blocking. A tabu search approach. *Omega*, *35*(3), 302–311. https://doi.org/10.1016/j.omega.2005.07.004

Groover, M. P. (2016). *Automation, production systems, and computer-integrated manufacturing*. Pearson Education India.

Guezori, M., & Houacine, A. (2012). Hybrid Flow Shop Scheduling Problem Using Artificial Immune System. *International Journal of Intelligent Systems and Applications*, *4*(10), 82–88. https://doi.org/10.5815/ijisa.2012.10.09

Gupta, J. N. D. (1971). A functional heuristic algorithm for the flowshop scheduling problem. *Operational Research Quarterly*, 39–47.

Gupta, J. N. D., & Stafford, E. F. (2006). Flowshop scheduling research after five decades. *European Journal of Operational Research*, *169*(3), 699–711. https://doi.org/10.1016/j.ejor.2005.02.001

Gupta, Shubham, & Deep, K. (2020). Hybrid sine cosine artificial bee colony algorithm for global optimization and image segmentation. *Neural Computing and Applications*, *32*(13), 9521–9543. https://doi.org/10.1007/s00521-019-04465-6

Gupta, SK, Zanakis, S., & Mandokovic, T. (1988). *POMS: Production and operations management software*.

Hadidi, A., Azad, S. K., & Azad, S. K. (2010). Structural optimization using artificial bee colony algorithm. *2nd International Conference on Engineering Optimization*.

Hakli, H., & Kiran, M. S. (2020). An improved artificial bee colony algorithm for balancing local and global search behaviors in continuous optimization. *International Journal of Machine Learning and Cybernetics*, 1–26.

Han, Y.-Y., Duan, J.-H., Yang, Y.-J., Zhang, M., & Yun, B. (2012). Minimizing the Total Flowtime Flowshop with Blocking Using a Discrete Artificial Bee Colony. In D.S. Huang, Y. Gan, P. Gupta & M. M. Gromiha (Eds.) *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence* (pp. 91-97). Zhengzhou, China: Springer.

Han, Y.-Y., Duan, J.-H., & Zhang, M. (2011). Apply the discrete artificial bee colony algorithm to the blocking flow shop problem with makespan criterion. *2011 Chinese Control and Decision Conference (CCDC)*, 2131–2135. https://doi.org/10.1109/CCDC.2011.5968558

Han, Y., Duan, J., & Zhang, M. (2011). *Apply the discrete artificial bee colony algorithm to the blocking flow shop problem with makespan criterion*. *252059*, 2131–2135.

Han, Y. Y., Pan, Q. K., Li, J. Q., & Sang, H. Y. (2012). An improved artificial bee colony algorithm for the blocking flowshop scheduling problem. *International Journal of Advanced Manufacturing Technology*, *60*(9–12), 1149–1159. https://doi.org/10.1007/s00170-011-3680-0

He, Q. M., & Alfa, A. S. (1998). The MMAP[K]/PH[K]/1 queues with a last-come-first-served preemptive service discipline. *Queueing systems, 29*(2-4), 269-291.

Holland, P. C., & Rescorla, R. A. (1975). The effect of two ways of devaluing the unconditioned stimulus after first- and second-order appetitive conditioning. *Journal of Experimental Psychology: Animal Behavior Processes*, *1*(4), 355–363. https://doi.org/10.1037/0097-7403.1.4.355

Hong, T. P., Huang, P. Y., & Horng, G. (2006). Using the LPT and the Palmer Approaches to Solve Group Flexible Flow-shop Problems. *IJCSNS, 6*(3A), 98.

Huang, V. L., Qin, A. K., Suganthan, P. N., & Tasgetiren, M. F. (2007). Multi-objective optimization based on self-adaptive differential evolution algorithm. *2007 IEEE Congress on Evolutionary Computation*, 3601–3608. https://doi.org/10.1109/CEC.2007.4424939

Hundal, T. S., & Rajgopal, J. (1988). An extension of Palmer's heuristic for the flow shop scheduling problem. *International Journal of Production Research, 26*(6), 1119-1124.

Hui, C. W., & Gupta, A. (2001). A bi-index continuous-time mixed-integer linear programming model for single-stage batch scheduling with parallel units. *Industrial and Engineering Chemistry Research*, *40*(25), 5960–5967. https://doi.org/10.1021/ie9909111

Janak, S. L., Floudas, C. A., Kallrath, J., & Vormbrock, N. (2006). Production scheduling of a large-scale industrial batch plant. I. Short-term and medium-term scheduling. *Industrial and Engineering Chemistry Research*, *45*(25), 8234–8252. https://doi.org/10.1021/ie0600588

Jemni, M., & Ladhari, T. (2011). Solving the Permutation Flow Shop Problem with Makespan Criterion using Grids. *International Journal of Grid and Distributed Computing*, *4*(2), 53–64.

Jolliffe, F. R., Gardner, M. J., & Altman, D. G. (1990). Statistics with Confidence: Confidence Intervals and Statistical Guidelines. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, *153*(2), 256. https://doi.org/10.2307/2982808

Johnson, S. M. (1954). Optimal two-and three-stage production schedules with setup times included. *Naval research logistics quarterly, 1*(1), 61-68.

Jouini, O. (2012). Analysis of a last come first served queueing system with customer abandonment. *Computers & Operations Research, 39*(12), 3040-3045.

Kahraman, C., Engin, O., Kaya, İ., & Kerim Yilmaz, M. (2008). An application of effective genetic algorithms for solving hybrid flow shop scheduling problems.

International Journal of Computational Intelligence Systems, *1*(2), 134–147. https://doi.org/10.1080/18756891.2008.9727611

Kalczynski, P. J., & Kamburowski, J. (2008). An improved NEH heuristic to minimize makespan in permutation flow shops. *Computers and Operations Research*, *35*(9), 3001–3008. https://doi.org/10.1016/j.cor.2007.01.020

Kan, A. H. G. R., & Kan, A. H. G. R. (2011). General flow-shop and job-shop problems. In *Machine Scheduling Problems* (pp. 106–130). https://doi.org/10.1007/978-1-4613-4383-7_6

Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. *Techn. Rep. TR06, Erciyes Univ. Press, Erciyes*.

Karaboga, D, & Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing Journal*, *8*(1), 687–697. https://doi.org/10.1016/j.asoc.2007.05.007

Karaboga, Dervis, & Akay, B. (2009). A comparative study of Artificial Bee Colony algorithm. *Applied Mathematics and Computation*, *214*(1), 108–132. https://doi.org/10.1016/j.amc.2009.03.090

Karaboga, Dervis, & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, *39*(3), 459–471. https://doi.org/10.1007/s10898-007-9149-x

Karaboga, Dervis, Gorkemli, B., Ozturk, C., & Karaboga, N. (2012). A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artificial Intelligence Review*, *42*(1), 21–57. https://doi.org/10.1007/s10462-012-9328-0

Karaboga, Dervis, Ozturk, C., Karaboga, N., & Gorkemli, B. (2012). Artificial bee colony programming for symbolic regression. *Information Sciences*, *209*, 1–15. https://doi.org/10.1016/J.INS.2012.05.002

Katragjini, K., Vallada, E., & Ruiz, R. (2013). Flow shop rescheduling under different types of disruption. *International Journal of Production Research*, *51*(3), 780–797. https://doi.org/10.1080/00207543.2012.666856

Kerzner, H. R. (2013). *Project management: a systems approach to planning, scheduling, and controlling*. John Wiley & Sons.

Khorramizadeh, M., & Riahi, V. (2015). A Bee Colony Optimization Approach for Mixed Blocking Constraints Flow Shop Scheduling Problems. *Mathematical Problems in Engineering*, *2015*. https://doi.org/10.1155/2015/612604

Klein, C. E., Bittencourt, M., & Coelho, L. dos S. (2015). Wavenet using artificial bee colony applied to modeling of truck engine powertrain components. *Engineering Applications of Artificial Intelligence*, *41*, 41–55. https://doi.org/http://dx.doi.org/10.1016/j.engappai.2015.01.009

Kouki, S., Guenaoui, M., & Jemni, M. (2016). A genetic algorithm for the permutation flow shop-problem: A parametric study. *2016 International Symposium on Networks, Computers and Communications, ISNCC 2016*, 1–6. https://doi.org/10.1109/ISNCC.2016.7746115

Kramer, O. (2017). Genetic Algorithm Essentials. In *Springer International Publishing AG* (Vol. 679). https://doi.org/10.1007/978-3-319-52156-5

Kulscar, G., & Forrai, M. K. (2009). Solving Multi-Objective Production Scheduling Problems Using A New Approach. *Production Systems and Information*

Engineering, *5*, 81–94.

Kumar, M., Husain, M., Upreti, N., & Gupta, D. (2020). Genetic Algorithm: Review and Application. *International Journal of Information Technology and Knowledge Management, 2*(2), 451-454. https://doi.org/10.2139/ssrn.3529843

Kurdi, M. (2020). A memetic algorithm with novel semi-constructive evolution operators for permutation flowshop scheduling problem. *Applied Soft Computing Journal*, *94*(2020). https://doi.org/10.1016/j.asoc.2020.106458

Lamini, C., Benhlima, S., & Elbekri, A. (2018). Genetic algorithm based approach for autonomous mobile robot path planning. *Procedia Computer Science*, *127*, 180–189. https://doi.org/10.1016/j.procs.2018.01.113

Lapierre, S. D., & Ruiz, A. B. (2007). Scheduling logistic activities to improve hospital supply systems. *Computers & Operations Research*, *34*(3), 624–641.

Laudis, L. L., Shyam, S., Suresh, V., & Kumar, A. (2018). A study: Various NP-hard problems in VLSI and the need for biologically inspired heuristics. *Advances in Intelligent Systems and Computing*, *708*, 193–204. https://doi.org/10.1007/978-981-10-8636-6_21

Lee, C. Y. (1997). Minimizing the makespan in the two-machine flowshop scheduling problem with an availability constraint. *Operations research letters, 20*(3), 129-139.

Lee, W.-P., & Cai, W.-T. (2011). A novel artificial bee colony algorithm with diversity strategy. *2011 Seventh International Conference on Natural Computation*, 1441–1444. https://doi.org/10.1109/ICNC.2011.6022505

Lee, Y. Z., & Ponnambalam, S. G. (2012). Optimisation of multipass turning operations using PSO and GA-AIS algorithms. *International Journal of Production Research*, *50*(22), 6499–6518. https://doi.org/10.1080/00207543.2011.653450

Leitão, P. (2009). Agent-based distributed manufacturing control: A state-of-the-art survey. *Engineering Applications of Artificial Intelligence*, *22*(7), 979–991.

Leonardi, S., & Raz, D. (2007). Approximating total flow time on parallel machines. *Journal of Computer and System Sciences*, *73*(6), 875–891. https://doi.org/10.1016/j.jcss.2006.10.018

Li, J. Q., Pan, Q. K., & Duan, P. Y. (2016). An Improved Artificial Bee Colony Algorithm for Solving Hybrid Flexible Flowshop with Dynamic Operation Skipping. *IEEE Transactions on Cybernetics*, *46*(6), 1311–1324. https://doi.org/10.1109/TCYB.2015.2444383

Li, J. Q., Pan, Q. K., & Mao, K. (2014). Hybrid particle swarm optimization for hybrid flowshop scheduling problem with maintenance activities. *Scientific World Journal*, *2014*. https://doi.org/10.1155/2014/596850

Li, J, Pan, Q., Xie, S., & Wang, S. (2011). *A Hybrid Artificial Bee Colony Algorithm for Flexible Job Shop Scheduling Problems Artificial Bee Colony Algorithm*. *VI*(2), 286–296.

Li, Jun-qing, & Pan, Q. (2015). Solving the large-scale hybrid flow shop scheduling problem with limited buffers by a hybrid artificial bee colony algorithm. *Information Sciences*, *316*, 487–502. https://doi.org/http://dx.doi.org/10.1016/j.ins.2014.10.009

Li, M., Du, W., & Nian, F. (2014). An adaptive particle swarm optimization algorithm based on directed weighted complex network. *Mathematical Problems in Engineering*, *2014*. https://doi.org/10.1155/2014/434972

Lian, Z., Gu, X., & Jiao, B. (2008). A novel particle swarm optimization algorithm for permutation flow-shop scheduling to minimize makespan. *Chaos, Solitons and Fractals*, *35*(5), 851–861. https://doi.org/10.1016/j.chaos.2006.05.082

Liao, C. J., Chao-Tang Tseng, & Luarn, P. (2007). A discrete version of particle swarm optimization for flowshop scheduling problems. *Computers and Operations Research*, *34*(10), 3099–3111. https://doi.org/10.1016/j.cor.2005.11.017

Liel, O., & Thirer, N. S. (2019). A rapid convergent genetic algorithm for NP-hard problems. In T. Pham (Ed.), *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, *11006* (2019). https://doi.org/10.1117/12.2516766

Lin, S.-W., & Ying, K.-C. (2014). ABC-based manufacturing scheduling for unrelated parallel machines with machine-dependent and job sequence-dependent setup times. *Computers & Operations Research*, *51*, 172–181. https://doi.org/http://dx.doi.org/10.1016/j.cor.2014.05.013

Lin, S. W., & Ying, K. C. (2013). Minimizing makespan and total flowtime in permutation flowshops by a bi-objective multi-start simulated-annealing algorithm. *Computers and Operations Research*, *40*(6), 1625–1647. https://doi.org/10.1016/j.cor.2011.08.009

Liu, B., Wang, L., & Jin, Y. H. (2008). An effective hybrid PSO-based algorithm for flow shop scheduling with limited buffers. *Computers & Operations Research, 35*(9), 2791-2806.

Liu, D., Jin, D., Baquero, C., He, D., Yang, B., & Yu, Q. (2013). Genetic Algorithm with a Local Search Strategy for Discovering Communities in Complex Networks. *International Journal of Computational Intelligence Systems*, *6*(2), 354–369. https://doi.org/10.1080/18756891.2013.773175

Liu, W., Jin, Y., & Price, M. (2017). A new improved NEH heuristic for permutation flowshop scheduling problems. *International Journal of Production Economics*, *193*, 21–30. https://doi.org/10.1016/J.IJPE.2017.06.026

Liu, Y. F., & Liu, S. Y. (2013). A hybrid discrete artificial bee colony algorithm for permutation flowshop scheduling problem. *Applied Soft Computing Journal*, *13*(3), 1459–1463. https://doi.org/10.1016/j.asoc.2011.10.024

Loukil, T., Teghem, J., & Tuyttens, D. (2005). Solving multi-objective production scheduling problems using metaheuristics. *European Journal of Operational Research*, *161*(1), 42–61. https://doi.org/10.1016/j.ejor.2003.08.029

Luo, J., Wang, Q., & Xiao, X. (2013). A modified artificial bee colony algorithm based on converge-onlookers approach for global optimization. *Applied Mathematics and Computation*, *219*(20), 10253–10262. https://doi.org/10.1016/j.amc.2013.04.001

Mahapatra, S., Dash, R. R., & Pradhan, S. K. (2017). Heuristics Techniques for Scheduling Problems with Reducing Waiting Time Variance. In *Heuristics and Hyper-Heuristics - Principles and Applications*. https://doi.org/10.5772/intechopen.69224

Marinakis, Y., Marinaki, M., & Matsatsinis, N. (2009). A hybrid discrete Artificial Bee Colony - GRASP algorithm for clustering. *2009 International Conference on*

Computers & Industrial Engineering, 548–553. https://doi.org/10.1109/ICCIE.2009.5223810

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115–133. https://doi.org/10.1007/BF02478259

McMahon, G., & Florian, M. (1975). On Scheduling With Ready Times And Due Dates To Minimize Maximum Lateness. *Operations Research*, *23*(3), 475–482. https://doi.org/10.1287/opre.23.3.475

Mirabi, M. (2011). Ant colony optimization technique for the sequence-dependent flowshop scheduling problem. *International Journal of Advanced Manufacturing Technology*, *55*(1–4), 317–326. https://doi.org/10.1007/s00170-010-3037-0

Moccellin, J. V. (1995). A new heuristic method for the permutation flow shop scheduling problem. *Journal of the Operational Research Society*, *46*(7), 883–886.

Naderi, B., & Ruiz, R. (2010). The distributed permutation flowshop scheduling problem. *Computers & Operations Research, 37*(4), 754-768.

Naderi, B., & Ruiz, R. (2014). A scatter search algorithm for the distributed permutation flowshop scheduling problem. *European Journal of Operational Research*, *239*(2), 323–334. https://doi.org/10.1016/J.EJOR.2014.05.024

Nawaz, M., Enscore, E. E., & Ham, I. (1983a). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, *11*(1), 91–95. https://doi.org/10.1016/0305-0483(83)90088-9

Nawaz, M., Enscore, E. E., & Ham, I. (1983b). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, *11*(1), 91–95.

Nowicki, E., & Smutnicki, C. (1996). A fast tabu search algorithm for the permutation flow-shop problem. *European Journal of Operational Research*, *91*(1), 160–175. https://doi.org/10.1016/0377-2217(95)00037-2

Osman, I., & Potts, C. (1989). Simulated annealing for permutation flow-shop scheduling. *Omega*, *17*(6), 551–557. https://doi.org/10.1016/0305-0483(89)90059-5

Öztop, H., Tasgetiren, M. F., Eliiyi, D. T., Pan, Q. K., & Kandiller, L. (2020). An energy-efficient permutation flowshop scheduling problem. *Expert Systems with Applications*, *150*. https://doi.org/10.1016/j.eswa.2020.113279

Palmer, D. S. (1965). Sequencing jobs through a multi-stage process in the minimum total time—a quick method of obtaining a near optimum. *Journal of the Operational Research Society, 16*(1), 101-107.

Pan, Q.-K., Fatih Tasgetiren, M., Suganthan, P. N., & Chua, T. J. (2011). A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. *Information Sciences*, *181*(12), 2455–2468. https://doi.org/10.1016/J.INS.2009.12.025

Pan, Q. K., & Ruiz, R. (2012). Local search methods for the flowshop scheduling problem with flowtime minimization. *European Journal of Operational Research, 222*(1), 31-43.

Pan, Q. K., & Ruiz, R. (2013). A comprehensive review and evaluation of permutation flowshop heuristics to minimize flowtime. *Computers and Operations Research*, Vol. 40, pp. 117–128. https://doi.org/10.1016/j.cor.2012.05.018

Pan, Q. K., & Ruiz, R. (2014). An effective iterated greedy algorithm for the mixed no-idle permutation flowshop scheduling problem. *Omega, 44*, 41-50.

Pan, Q. K., Wang, L., Li, J. Q., & Duan, J. H. (2014). A novel discrete artificial bee colony algorithm for the hybrid flowshop scheduling problem with makespan minimisation. *Omega (United Kingdom)*, *45*, 42–56. https://doi.org/10.1016/j.omega.2013.12.004

Pan, Q. K., Tasgetiren, M. F., & Liang, Y. C. (2008). A discrete differential evolution algorithm for the permutation flowshop scheduling problem. *Computers & Industrial Engineering, 55*(4), 795-816.

Pan, Q., Tasgetiren, M. F., Suganthan, P. N., & Liang, Y. (2010). Solving Lot-streaming Flow Shop Scheduling Problems Using A Discrete Harmony Search Algorithm. *IEEE World Congress on Computational Intelligence*, 18–23. https://doi-org.ezproxy.uthm.edu.my/10.1109/CEC.2010.5586199

Pansuwan, P., Rukwong, N., & Pongcharoen, P. (2010). Identifying Optimum Artificial Bee Colony (ABC) Algorithm's Parameters for Scheduling the Manufacture and Assembly of Complex Products. *2010 Second International Conference on Computer and Network Technology*, 339–343. https://doi.org/10.1109/ICCNT.2010.25

Park, S. H. (1976). Scheduling Theory And Problems: Review And Categorization Of Solution Procedured. *Journal of Korean Institute of Industrial Engineers*, *2*(1), 101–108.

Pasupathy, T., Rajendran, C., & Suresh, R. K. (2006). A multi-objective genetic algorithm for scheduling in flow shops to minimize the makespan and total flow time of jobs. *International Journal of Advanced Manufacturing Technology*, *27*(7–8), 804–815. https://doi.org/10.1007/s00170-004-2249-6

Paternina-Arboleda, C. D., Montoya-Torres, J. R., Acero-Dominguez, M. J., & Herrera-Hernandez, M. C. (2008). Scheduling jobs on a k-stage flexible flow-shop. *Annals of Operations Research*, *164*(1), 29–40. https://doi.org/10.1007/s10479-007-0257-2

Pauzi, N. F. B. M., & Bareduan, S. A. (2016). Scheduling analysis for flowship using artifical bee colony (ABC) algorithm with varying onlooker approaches. *ARPN Journal of Engineering and Applied Sciences*, *11*(10), 6472–6477.

Pinedo, M. L. (2012). *Scheduling: theory, algorithms, and systems*. Springer Science & Business Media.

Qian, B., Wang, L., Huang, D. X., & Wang, X. (2009). Multi-objective no-wait flow-shop scheduling with a memetic algorithm based on differential evolution. *Soft Computing, 13*(8-9), 847.

Qian, B., Wang, L., Huang, D. X., Wang, W. L., & Wang, X. (2009). An effective hybrid DE-based algorithm for multi-objective flow shop scheduling with limited buffers. *Computers & Operations Research, 36*(1), 209-233.

Quadt, D., & Kuhn, H. (2007). A taxonomy of flexible flow line scheduling procedures. *European Journal of Operational Research*, *178*(3), 686–698. https://doi.org/10.1016/j.ejor.2006.01.042

Rabanimotlagh, A. (2011). An efficient ant colony optimization algorithm for multiobjective flow shop scheduling problem. *World Academy of Science, Engineering and Technology*, *75*(54), 127–133.

Radha Ramanan, T., Iqbal, M., & Umarali, K. (2014). A particle swarm optimization approach for permutation flow shop scheduling problem. *International Journal for Simulation and Multidisciplinary Design Optimization*, *5*, A20. https://doi.org/10.1051/smdo/2013006

Rajendran, C., & Ziegler, H. (2004). Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs. *European Journal of Operational Research*, *155*(2), 426–438. https://doi.org/10.1016/S0377-2217(02)00908-6

Rajendran, C., & Ziegler, H. (2005). Two ant-colony algorithms for minimizing total flowtime in permutation flowshops. *Computers and Industrial Engineering*, *48*(4), 789–797. https://doi.org/10.1016/j.cie.2004.12.009

Ravindran, D., Selvakumar, S. J., Sivaraman, R., & Haq, A. N. (2005). Flow shop scheduling with multiple objective of minimizing makespan and total flow time. *International Journal of Advanced Manufacturing Technology*, *25*(9–10), 1007–1012. https://doi.org/10.1007/s00170-003-1926-1

Reeves, C.R. (1995). A genetic algorithm for flow shop sequencing. *Computers & Operations Research*, *22*(1), 5–11.

Riahi, V., Chiong, R., & Zhang, Y. (2020). A new iterated greedy algorithm for no-idle permutation flowshop scheduling with the total tardiness criterion. *Computers & Operations Research, 117*, 104839.

Ribas, I., Companys, R., & Tort-Martorell, X. (2011). An iterated greedy algorithm for the flowshop scheduling problem with blocking. *Omega*, *39*(3), 293–301. https://doi.org/10.1016/j.omega.2010.07.007

Roberge, V., Tarbouchi, M., & Labonte, G. (2013). Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. *IEEE Transactions on Industrial Informatics*, *9*(1), 132–141. https://doi.org/10.1109/TII.2012.2198665

Roychowdhury, S., Allen, T. T., & Allen, N. B. (2017). A genetic algorithm with an earliest due date encoding for scheduling automotive stamping operations. *Computers & Industrial Engineering, 105*, 201-209.

Ruiz, R., & Maroto, C. (2005). A comprehensive review and evaluation of permutation flowshop heuristics. *European Journal of Operational Research*, *165*(2), 479–494. https://doi.org/http://dx.doi.org/10.1016/j.ejor.2004.04.017

Ruiz, R., Maroto, C., & Alcaraz, J. (2005). Solving the flowshop scheduling problem with sequence dependent setup times using advanced metaheuristics. *European Journal of Operational Research*, *165*(1), 34-54. https://doi.org/10.1016/j.ejor.2004.01.022.

Ruiz, R., Pan, Q. K., & Naderi, B. (2019). Iterated Greedy methods for the distributed permutation flowshop scheduling problem. *Omega (United Kingdom)*, *83*, 213–222. https://doi.org/10.1016/j.omega.2018.03.004

Ruiz, R., & Stützle, T. (2007). A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research*, *177*(3), 2033–2049. https://doi.org/10.1016/j.ejor.2005.12.009

Saravanan, M., Joseph Dominic Vijayakumar, S., & Srinivasan, R. (2014). A multicriteria permutation flowshop scheduling problem with setup times. *International Journal of Engineering and Technology*, *6*(3), 1329–1339.

Sauer, J. (1999). Knowledge-based systems techniques and applications in scheduling. *Knowledge-Based Systems Techniques and Applications, TL Leondes, Ed., San Diego, Academic Press*.

Selvi, V., & Umarani, D. R. (2010). Comparative Analysis of Ant Colony and Particle Swarm Optimization Techniques. *International Journal of Computer Applications*, *5*(4), 1–6. https://doi.org/10.5120/908-1286

Sha, D. Y., & Hung Lin, H. (2009). A particle swarm optimization for multi-objective flowshop scheduling. *International Journal of Advanced Manufacturing Technology*, *45*(7–8), 749–758. https://doi.org/10.1007/s00170-009-1970-6

Shyu, S. J., Lin, B. M. T., & Yin, P. Y. (2004). Application of ant colony optimization for no-wait flowshop scheduling problem to minimize the total completion time. *Computers and Industrial Engineering*, *47*(2–3), 181–193. https://doi.org/10.1016/j.cie.2004.06.006

Sidek, N. A., Bareduan, S. A., & Nawawi, A. (2019). Performance Investigation of Artificial Bee Colony (ABC) Algorithm for Permutation Flowshop Scheduling Problem (PFSP). *Journal of Physics: Conference Series*, *1150*(1), 012060. https://doi.org/10.1088/1742-6596/1150/1/012060

Sidek, N. A., & Bareduan, S. A. (2014). Effect of Trial Counter Variation on the Performance of Artificial Bee Colony (ABC) Algorithm for Permutation Flowshop Scheduling Problem (PFSP). *International Conference on Mathematics, Engineering and Industrial Applications (ICoMEIA 2014)*. Penang, Malaysia: Institute of Engineering Mathematics, Universiti Malaysia Perlis (UniMAP).

Singhal, E., Singh, S., & Dayma, A. (2012). An Improved Heuristic for Permutation Flow Shop Scheduling (NEH ALGORITHM). *International Journal Of Computational Engineering Research (Ijceronline.Com)*, *2*(6), 2250–3005.

Sonmez, M. (2011). Artificial Bee Colony algorithm for optimization of truss structures. *Applied Soft Computing*, *11*(2), 2406–2418. https://doi.org/10.1016/J.ASOC.2010.09.003

Sulaiman, N., Mohamad-Saleh, J., & Ghani Abro, A. (2013). A Modified Artificial Bee Colony (JA-ABC) Optimization Algorithm. *Proceedings of the 2013 International Conference on Applied Mathematics and Computational Methods in Engineering*, 74–79.

Taillard, E. (1993). Benchmarks for basic scheduling problems. *European Journal of Operational Research*, *64*(2), 278–285. https://doi.org/10.1016/0377-2217(93)90182-M

Taillard, E. (1990). Some efficient heuristic methods for the flow shop sequencing problem. *European Journal of Operational Research*, *47*(1), 65–74.

Talbi, E. G. (2009). *Metaheuristics: From Design to Implementation*. Hoboken, New Jersey: John Wiley & Sons.

Tasgetiren, M. F., Kizilay, D., Pan, Q. K., & Suganthan, P. N. (2017). Iterated greedy algorithms for the blocking flowshop scheduling problem with makespan criterion. *Computers & Operations Research, 77*, 111-126.

Tasgetiren, M. F., Liang, Y.-C., Sevkli, M., & Gencyilmaz, G. (2007). A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem. *European Journal of Operational*

Research, *177*(3), 1930–1947. https://doi.org/10.1016/J.EJOR.2005.12.024

Tasgetiren, M. F., Pan, Q. K., Suganthan, P. N., & Buyukdagli, O. (2013). A variable iterated greedy algorithm with differential evolution for the no-idle permutation flowshop scheduling problem. *Computers & Operations Research, 40*(7), 1729-1743.

Tasgetiren, M. F., Pan, Q.-K., Suganthan, P. N., & Chen, A. H. L. (2010). A discrete artificial bee colony algorithm for the permutation flow shop scheduling problem with total flowtime criterion. *IEEE Congress on Evolutionary Computation*, 1–8. IEEE.

Tasgetiren, M. F., Pan, Q. K., Suganthan, P. N., & Chen, A. H. (2011). A discrete artificial bee colony algorithm for the total flowtime minimization in permutation flow shops. *Information Sciences*, *181*(16), 3459–3475. https://doi.org/10.1016/j.ins.2011.04.018

Tasgetiren, M. F., Pan, Q., Suganthan, P. N., & Chen, A. H. (2010). A Discrete Artificial Bee Colony Algorithm for the Permutation Flow. *IEEE World Congress on Computational Intelligence*, *181*(16), 18–23. https://doi.org/http://dx.doi.org/10.1016/j.ins.2011.04.018

Tasgetiren, M. F., Sevkli, M., Liang, Y. C., & Gencyilmaz, G. (2004). Particle swarm optimization algorithm for permutation flowshop sequencing problem. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *3172 LNCS*, 382–389. https://doi.org/10.1007/978-3-540-28646-2_38

Tasgetiren, M. F., Yüksel, D., Gao, L., Pan, Q. K., & Li, P. (2019). A Discrete Artificial Bee Colony Algorithm for the Energy-Efficient No-Wait Flowshop Scheduling Problem. *Procedia Manufacturing, 39*, 1223-1231.

Tavares Neto, R. F., & Godinho Filho, M. (2011). An ant colony optimization approach to a permutational flowshop scheduling problem with outsourcing allowed. *Computers & Operations Research*, *38*(9), 1286–1293. https://doi.org/10.1016/J.COR.2010.11.010

Toutouh, J. (2015). *Natural Computing for Vehicular Networks* (Doctoral dissertation ,Universidad De Malaga, Spain). Retrieved from https://riuma.uma.es/xmlui/handle/10630/13432

Trelea, I. C. (2003). The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information Processing Letters*, *85*(6), 317–325. https://doi.org/http://dx.doi.org/10.1016/S0020-0190(02)00447-7

Tsai, C. F., & Lu, S. L. (2016). A Novel Mechanism for Efficient the Search Optimization of Genetic Algorithm. *International Journal of Computational Intelligence Systems*, *9*(1), 57–64. https://doi.org/10.1080/18756891.2016.1144153

Tsai, H.-C. (2014). Integrating the artificial bee colony and bees algorithm to face constrained optimization problems. *Information Sciences*, *258*, 80–93. https://doi.org/http://dx.doi.org/10.1016/j.ins.2013.09.015

Tseng, C. T., & Liao, C. J. (2008). A particle swarm optimization algorithm for hybrid flow-shop scheduling with multiprocessor tasks. *International Journal of Production Research*, *46*(17), 4655–4670. https://doi.org/10.1080/00207540701294627

Tseng, L. Y., & Lin, Y. T. (2006). A hybrid genetic algorithm for the flow-shop scheduling problem. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 4031 LNAI*, 218–227. https://doi.org/10.1007/11779568_25

Tseng, L. Y., & Lin, Y. T. (2009). A hybrid genetic local search algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research*, *198*(1), 84–92. https://doi.org/10.1016/j.ejor.2008.08.023

Uzsoy, R., Lee, C. -Y, & Martin-Vega, L. A. (1992). Scheduling semiconductor test operations: Minimizing maximum lateness and number of tardy jobs on a single machine. *Naval Research Logistics (NRL)*, *39*(3), 369–388. https://doi.org/10.1002/1520-6750(199204)39:3<369::AID-NAV3220390307>3.0.CO;2-F

Vollmann, T. E., William, L. B., Whybark, D. C., & Jacobs, F. R. (2005). Manufacturing Planning and Control for Supply Chain Management. In *McGraw-Hill/Irwin. ISBN*. Retrieved from http://books.mcgraw-hill.com/cgi-bin/pbg/0072299908.html

Wang, Guangchen, Gao, L., Li, X., Li, P., & Tasgetiren, M. F. (2020). Energy-efficient distributed permutation flow shop scheduling problem using a multi-objective whale swarm algorithm. *Swarm and Evolutionary Computation*, *57*(2020). https://doi.org/10.1016/j.swevo.2020.100716

Wang, Guilong, Zhao, G., & Wang, X. (2014). Development and evaluation of a new rapid mold heating and cooling method for rapid heat cycle molding. *International Journal of Heat and Mass Transfer, 78*(2014), 99-111. https://doi.org/10.1016/j.ijheatmasstransfer.2014.06.062

Wang, L., Pan, Q. K., Suganthan, P. N., Wang, W. H., & Wang, Y. M. (2010). A novel hybrid discrete differential evolution algorithm for blocking flow shop scheduling problems. *Computers and Operations Research*, *37*(3), 509–520. https://doi.org/10.1016/j.cor.2008.12.004

Wang, R., Shi, K., & Wang, S. (2020). Research on Performance of Artificial Bee Colony Algorithm Based on Benchmark Test Function. *MS&E*, *740*(1), 12019.

Wang, Y. M., & Yin, H. L. (2013). A two-stage approach based on genetic algorithm for large size flow shop scheduling problem. *2013 IEEE International Conference on Mechatronics and Automation, IEEE ICMA 2013*, 376–381. https://doi.org/10.1109/ICMA.2013.6617948

Wang, L., Zhang, L., & Zheng, D. Z. (2006). An effective hybrid genetic algorithm for flow shop scheduling with limited buffers. *Computers & Operations Research, 33*(10), 2960-2971.

Woeginger, G. J. (2003a). Exact algorithms for NP-hard problems: A survey. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *2570*, 185–207. https://doi.org/10.1007/3-540-36478-1_17

Wolde, M. G., Berhan, E., & Jilcha, K. (2018). Production improvement with flow shop scheduling heuristics in Household utensils manufacturing company. *Cogent Engineering*, *5*(1). https://doi.org/10.1080/23311916.2018.1430007

Wong, L. P., Puan, C. Y., Low, M., & Chong, C. S. (2008). Bee colony optimization algorithm with big valley landscape exploitation for job shop scheduling problems. *International Journal of Bio-Inspired Computation, 2*(2), 85-99.

Xoxa, N., Zotaj, M., Tafa, I., & Fejzaj, J. (2014). Simulation of first come first served (FCFS) and shortest job first (SJF) algorithms. *Tirana, Albania: IJCSN-International Journal of Computer Science and Network, 3*(6), 444-449.

Xu, C., Xu, J., & Chang, H. (2009). Ant colony optimization based on estimation of distribution for the traveling salesman problem. *5th International Conference on Natural Computation, ICNC 2009*, *3*, 19–23. https://doi.org/10.1109/ICNC.2009.165

Yagmahan, B., & Yenisey, M. M. (2008). Ant colony optimization for multi-objective flow shop scheduling problem. *Computers and Industrial Engineering*, *54*(3), 411–420. https://doi.org/10.1016/j.cie.2007.08.003

Yagmahan, B., & Yenisey, M. M. (2010). A multi-objective ant colony system algorithm for flow shop scheduling problem. *Expert Systems with Applications*, *37*(2), 1361–1368. https://doi.org/10.1016/j.eswa.2009.06.105

Yalaoui, F., & Chu, C. (2002). Parallel machine scheduling to minimize total tardiness. *International Journal of Production Economics*, *76*(3), 265–279. https://doi.org/10.1016/S0925-5273(01)00175-X

Yang, X. (2008). *Introduction to Mathematical Optimization: From Linear Programming to Metaheuristics*. England: Cambridge International Science Publishing.

Yi, L. (2016). Study on an improved PSO algorithm and its application for solving function problem. *International Journal of Smart Home*, *10*(3), 51–62. https://doi.org/10.14257/ijsh.2016.10.3.06

Yuan, K., Hennequin, S., Wang, X., & Gao, L. (2006). A new heuristic-EM for permutation flowshop scheduling. *IFAC Proceedings Volumes*, *39*(3), 33-38.

Yüksel, D., Taşgetiren, M. F., Kandiller, L., & Gao, L. (2020). An energy-efficient bi-objective no-wait permutation flowshop scheduling problem to minimize total tardiness and total energy consumption. *Computers & Industrial Engineering*, 106431.

Zhang, G., & Xing, K. (2019). Differential evolution metaheuristics for distributed limited-buffer flowshop scheduling with makespan criterion. *Computers and Operations Research*, *108*, 33–43. https://doi.org/10.1016/j.cor.2019.04.002

Zhang, L., Xiao, C., & Fei, T. (2017). Improved ant colony optimization algorithm based on RNA computing. *Automatic Control and Computer Sciences*, *51*(5), 366–375. https://doi.org/10.3103/S0146411617050108

Zhang, P. (2020). Research on Convergence of Artificial bee colony algorithm based on crossover and consistency distribution-good point set. *IOP Conference Series: Earth and Environmental Science*, *446*(5), 52007. https://doi.org/10.1088/1755-1315/446/5/052007

Zhang, S. J., & Gu, X. S. (2015). An effective discrete artificial bee colony algorithm for flow shop scheduling problem with intermediate buffers. *Journal of Central South University*, *22*(9), 3471–3484. https://doi.org/10.1007/s11771-015-2887-x

Zhao, F., Zhang, L., Zhang, Y., Ma, W., Zhang, C., & Song, H. (2020). A hybrid discrete water wave optimization algorithm for the no-idle flowshop scheduling problem with total tardiness criterion. *Expert Systems with Applications, 146*, 113166.

Zhong, J. H., & Zhang, J. (2012). Ant colony optimization algorithm for lifetime maximization in wireless sensor network with mobile sink. *GECCO'12 - Proceedings of the 14th International Conference on Genetic and Evolutionary Computation*, 1199–1204. https://doi.org/10.1145/2330163.2330328

Zhou, H., Feng, Y., & Han, L. (2001). The hybrid heuristic genetic algorithm for job shop scheduling. *Computers & Industrial Engineering, 40*(3), 191-200.

Zhou, R., Lee, H. P., & Nee, A. Y. C. (2008). Applying Ant Colony Optimisation (ACO) algorithm to dynamic job shop scheduling problems. *International Journal of Manufacturing Research*, *3*(3), 301–320. https://doi.org/10.1504/IJMR.2008.019212

Zhu, Z., & Heady, R. B. (2000). Minimizing the sum of earliness/tardiness in multi-machine scheduling: a mixed integer programming approach. *Computers and Industrial Engineering*, *38*(2), 297–305. https://doi.org/10.1016/S0360-8352(00)00048-6